Partitioned Quasi-Newton Methods for Nonlinear Equality Constrained Optimization

Tom Coleman*
Peter A. Fenyes

TR 88-931
August 1988

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

* Research partially supported by the Applied Mathematical Sciences Research Program (KC-04-02) of the Office of Energy Research of the U. S. Department of Energy under grant DE-FG02-86ER25013.A000 and by the U. S. Army Research Office through the Mathematical Sciences Institute, Cornell University.

PARTITIONED QUASI-NEWTON METHODS FOR NONLINEAR EQUALITY CONSTRAINED OPTIMIZATION*

THOMAS F. COLEMAN† AND PETER A. FENYES‡

Abstract. We derive new quasi-Newton updates for the (nonlinear) equality constrained minimization problem. The new updates satisfy a quasi-Newton equation, maintain positive definiteness on the null space of the active constraint matrix, and satisfy a minimum change condition. The application of the updates is not restricted to a small neighbourhood of the solution. In addition to derivation and motivational remarks, we discuss various numerical subtleties and provide results of numerical experiments.

Key Words. constrained optimization, equality constraints, numerical optimization, quasi-Newton method, secant method, sequential quadratic programming, SQP-method,augmented Lagrangian method, penalty function methods.

Subject Classification. AMS/MOS: 65H10, 65K05, 65K10.

Abbreviated Title. partitioned quasi-Newton methods

* Written August 1988.

† Computer Science Department and Center for Applied Mathematics, Cornell University, Ithaca, NY 14853. Research partially supported by the Applied Mathematical Sciences Research Program (KC-04-02) of the Office of Energy Research of the U.S. Department of Energy under grant DE-FG02-86ER25013.A000 and by the U.S. Army Research Office through the Mathematical Sciences Institute, Cornell University.

‡ General Motors Research Laboratories, Warren, Michigan

1. The Nonlinear Equality Problem

In this paper we derive new quasi-Newton updates for the nonlinear equality constrained problem

$$\min_x f(x) \quad \text{subject to} \quad c(x) = 0 \quad . \tag{NEP}$$

where $x \in R^n$, $f: R^n \rightarrow R$ and $c: R^n \rightarrow R^t$ and $t \le n$. The functions f and c_i, $i=1,\ldots,t$ are twice continuously differentiable. The new updates satisfy a quasi-Newton equation, maintain positive definiteness on the null space of the matrix of active gradients, and satisfy a minimum change condition. Furthermore, the application of the new updates is not restricted to a small neighborhood near the solution. To further motivate the development of our updates we will first review optimality conditions and some existing update strategies.

It will be useful to define $A \in R^{n \times t}$ as the matrix whose columns are the gradients of the constraints. In other words $A(x)=[\nabla c_1(x), \cdots, \nabla c_t(x)]$. If A is of full column rank, we can define two matrices $Y \in R^{n \times t}$ and $Z \in R^{n \times (n-t)}$ using the QR factorization

$$A = Q\bar{R} = [Y\ Z] \begin{bmatrix} R \\ 0 \end{bmatrix}$$

where $Q \doteq [Y\ Z]$ is orthogonal, $\bar{R}^T \doteq [R^T\ 0]$, and $R \in R^{t \times t}$ is upper triangular. When defined this way, Y and Z have orthonormal columns and satisfy the following relationships: $QQ^T = YY^T + ZZ^T = I_{n \times n}$, $Y^T Y = I_{t \times t}$, $Z^T Z = I_{m \times m}$, $Y^T Z = 0_{t \times m}$, and $Z^T Y = 0_{m \times t}$ where we have taken $m = n-t$. Thus, Y and Z represent orthogonal bases for the range of A and the null space of A^T respectively.

The solution to (NEP) must satisfy the first order necessary condition for optimality:

First Order Necessary Condition

If x^* is a local minimizer of (NEP), Z is defined as above, and a constraint qualification holds at x^*, then x^* satisfies

$$\begin{aligned} Z(x^*)^T \nabla_x f(x^*) &= 0 \\ c(x^*) &= 0 \quad . \end{aligned} \tag{1.1}$$

We may restate this in terms of the Lagrangian

$$L(x,\lambda) = f(x) - \sum_{i=1}^{t} \lambda_i c_i(x) \quad ,$$

with $\lambda \in R^t$. The first order necessary conditions for optimality then requires the existence of Lagrange multipliers λ^* such that x^* and λ^* satisfy

$$d_L(x^*,\lambda^*) = \begin{bmatrix} \nabla_x L(x^*,\lambda^*) \\ \nabla_\lambda L(x^*,\lambda^*) \end{bmatrix} = \begin{bmatrix} g(x^*) - A(x^*)\lambda(x^*) \\ c(x^*) \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix} \tag{1.2}$$

where $d_L \in R^{n+t}$ and we have defined $g \doteq \nabla_x f$ and $A\lambda = \sum \lambda_i \nabla_x c_i$.

In the remainder of this paper we will also assume that the solution (x^*,λ^*) satisfies the second order sufficiency condition. Specifically we assume:

- f and c_i, $i=1,\ldots,t$ are twice continuously differentiable,

- $\nabla c_i(x^*)$, i=1,...,t are linearly independent and thus satisfy a constraint qualification,
- $Z(x^*)^T \nabla_{xx} L(x^*,\lambda^*) Z(x^*)$ is positive definite where $\nabla_{xx} L(x,\lambda)$ is the second derivative, or Hessian matrix, of the Lagrangain.

Note that although many authors make the strong assumption that the full Hessian at (x^*,λ^*) is positive definite, we assume that only the reduced Hessian matrix, $Z(x^*)^T \nabla_{xx} L(x^*,\lambda^*) Z(x^*)$, is positive definite as required by the second order condition.

1.1. Iterative Methods for NEP - The Nonlinear Equations View

Let us first consider the solution of (NEP) in a general way, emphasizing the matrices we wish to approximate in a quasi-Newton approach. Most methods designed to solve (NEP) strive for a point x and a set of multipliers λ such that the first order necessary conditions hold. Thus we seek a solution (x^*,λ^*) to the set of $n+t$ nonlinear equations (1.2) in $n+t$ unknowns. Let us consider applying a Newton method to this system. The Jacobian of $d_L(x,\lambda)$ with respect to x and λ is given by

$$d_L{}'(x,\lambda) = \begin{bmatrix} \nabla_x d_L(x,\lambda)^T \\ \nabla_\lambda d_L(x,\lambda)^T \end{bmatrix} = \begin{bmatrix} \nabla_{xx} L(x,\lambda) & -A(x) \\ A(x)^T & 0 \end{bmatrix} .$$

The kth step of a Newton iteration is then obtained by solving

$$d_L{}'(x_k,\lambda_k) \begin{bmatrix} s_k \\ \gamma_k \end{bmatrix} = -d_L(x_k,\lambda_k) .$$

Dropping the subscript k, using the subscript + for $k+1$, and using the definitions of $s \doteq x_+ - x$ and $\gamma \doteq \lambda_+ - \lambda$ we can write the Newton iteration as

$$\begin{bmatrix} \nabla_{xx} L(x,\lambda) & -A(x) \\ A(x)^T & 0 \end{bmatrix} \begin{bmatrix} s \\ \lambda_+ \end{bmatrix} = - \begin{bmatrix} g \\ c \end{bmatrix} \tag{1.3}$$

If A is of full column rank and $\nabla_{xx} L(x,\lambda)$ is positive definite in the null-space of A^T then there exists a unique solution of the linear system (1.3). We can write the solution conveniently by using the orthogonal matrices Y and Z. Define

$$\overline{Q} \doteq \begin{bmatrix} Y & Z & 0 \\ 0 & 0 & I_{t \times t} \end{bmatrix}$$

so that $\overline{Q} \in R^{(n+t)\times(n+t)}$ and is an orthogonal matrix. Thus the solution to (1.3) is unchanged if we multiply both sides by $\overline{Q}^T$ and insert $\overline{Q}\overline{Q}^T = I$

$$\overline{Q}^T \begin{bmatrix} \nabla_{xx} L(x,\lambda) & -A(x) \\ A(x)^T & 0 \end{bmatrix} \overline{Q}\overline{Q}^T \begin{bmatrix} s \\ \lambda_+ \end{bmatrix} = -\overline{Q}^T \begin{bmatrix} g \\ c \end{bmatrix} .$$

Dropping the arguments x and λ and carrying out the multiplication by $\overline{Q}$ we have

$$\begin{bmatrix} Y^T \nabla^2 L Y & Y^T \nabla^2 L Z & -R \\ Z^T \nabla^2 L Y & Z^T \nabla^2 L Z & 0 \\ R^T & 0 & 0 \end{bmatrix} \begin{bmatrix} Y^T s \\ Z^T s \\ \lambda_+ \end{bmatrix} = - \begin{bmatrix} Y^T g \\ Z^T g \\ c \end{bmatrix} \tag{1.4}$$

where we use the shorthand $\nabla^2 L$ to denote the Hessian $\nabla_{xx} L(x,\lambda)$. This system is written using the four matrices $Y^T \nabla^2 L Y$, $Y^T \nabla^2 L Z$, $Z^T \nabla^2 L Y$, and $Z^T \nabla^2 L Z$ which we will refer to as reduced or projected Hessian matrices. Together, the four blocks will be referred to as the partitioned Hessian matrix, and we will call the system (1.4) the partitioned or transformed Newton system. Since this system is in block triangular form, we can immediately write down its solution. Defining

$$s_Y \doteq Y^T s \qquad g_Y \doteq Y^T g$$
$$s_Z \doteq Z^T s \qquad g_Z \doteq Z^T g$$

we have the Newton step

$$\begin{aligned} s &= Ys_Y + Zs_Z \quad . \\ s_Y &= -R^{-T}c \\ s_Z &= -(Z^T \nabla^2 LZ)^{-1}(g_Z + Z^T \nabla^2 LYs_Y) \\ \lambda_+ &= R^{-1}(g_Y + Y^T \nabla^2 LYs_Y + Y^T \nabla^2 LZs_Z) \quad . \end{aligned} \tag{1.5}$$

Note that we obtain the full step, s, as the sum of two orthogonal components, $v = Ys_Y$ and $h = Zs_Z$. Since Zs_Z lies in the null space of the constraints, the horizontal step, h, tends to decrease $Z^T g$ while moving tangent to the constraint contour. Similarly, since Ys_Y lies in the range space of the constraints, the vertical step, v, tends to reduce the constraint values.

Using equations (1.5), the Newton algorithm for (NEP) can be defined as follows:

NEWTON ALGORITHM

INITIALIZE:

$k \leftarrow 0$.

Choose x_0, λ_0.

DO WHILE ($||Z_k^T \nabla f_k|| > \varepsilon_{stop}$ or $||c_k|| > \varepsilon_{stop}$)

Evaluate f_k, ∇L_k, $\nabla^2 L_k$, c_k, A_k.

Factor $A_k = [Y_k \;\; Z_k][R_k^T \;\; 0^T]^T$.

SOLVE (1.5) for $s_Y{}^k$, $s_Z{}^k$, and λ_{k+1}.

$x_{k+1} \leftarrow x_k + v_k + h_k = x_k + Ys_Y{}^k + Zs_Z{}^k$

$k \leftarrow k+1$.

END

We mention that the system (1.3) can also be derived as a step in a Sequential Quadratic Programming (SQP) algorithm. Suppose at step k of the algorithm we are at the point (x_k, λ_k). The SQP algorithm solves the equality constrained quadratic program

$$\begin{aligned} \min_s \;\; & q_k = g(x_k)^T s + \tfrac{1}{2} s^T B_k s \\ \text{subject to} \;\; & c(x_k) + A(x_k)^T s = 0 \quad . \end{aligned} \tag{SQP}$$

If we take as B_k the exact Hessian, $\nabla^2 L(x_k, \lambda_k)$, then the first order conditions applied to (SQP) defines the system (1.3). If the matrix A is full rank and $\nabla^2 L(x, \lambda)$ is positive definite in the null space of A , the step s and the new Lagrange multipliers λ_+ defined by the solution to (SQP) are of course identical to the Newton step (1.5).

Next, we consider a quasi-Newton approach, assuming we can approximate the transformed second derivative matrix $[Y \; Z]^T \nabla^2 L \, [Y \; Z]$. In other words we approximate

$$\begin{bmatrix} B_{YY} & B_{YZ} \\ B_{ZY} & B_{ZZ} \end{bmatrix} \sim \begin{bmatrix} Y^T \nabla^2 LY & Y^T \nabla^2 LZ \\ Z^T \nabla^2 LY & Z^T \nabla^2 LZ \end{bmatrix} \quad . \tag{1.6}$$

Substituting our approximation into (1.4) and (1.5) we obtain: quasi-Newton SQP step

$$
\begin{aligned}
s &= Ys_Y + Zs_Z \quad . \\
s_Y &= -R^{-T}c \quad , \\
s_Z &= -B_{ZZ}^{-1}(Z^T g - B_{ZY}s_Y) \quad , \\
\lambda^+ = \lambda^{QP} &= R^{-1}(Y^T g + B_{YY}s_Y + B_{YZ}s_Z) \quad .
\end{aligned} \tag{1.7}
$$

In the remainder of this paper we will restrict our attention to quasi-Newton methods for (NEP). In the next section we will discuss the quasi-Newton methods currently used to solve (NEP) and show that they all lack certain desirable features. We remedy these deficiencies in Section 3 by deriving two new quasi-Newton updates with very desirable properties. In Section 4 we will analyze the formulas derived for the two new updates. Finally, we present the results of our extensive computational experience with the two new updates in Section 5.

2. Current Quasi-Newton Methods for the Nonlinear Equality Problem

Although quasi-Newton methods for (NEP) are quite complex, space does not permit a full discussion. Greater detail may be found in Fenyes (1987). Here, we will concentrate on what appears to be the major deficiency of current quasi-Newton methods for the constrained problem - simple, reliable methods for keeping the null-space Hessian approximation positive definite, particularly when far from the solution. In the unconstrained case, positivity of the Hessian approximation insures that the quasi-Newton step $s=-B^{-1}\nabla f$ will be a descent direction for the objective function at each iteration. In constrained methods, keeping at least B_{ZZ} positive definite is generally accepted to be desirable. Here, we prove that when B_{ZZ} is a positive definite approximation to $Z^T\nabla^2 LZ$, the SQP step (1.7) yields a well defined descent direction for the augmented Lagrangian merit function

$$L_a(x,\lambda,\sigma) \doteq f(x) - c(x)^T\lambda + \tfrac{1}{2}\sigma c(x)^T c(x) \quad .$$

We examine two choices for the Lagrange multipliers in the definition of L_a: $\lambda=\lambda^{QP}$ where λ^{QP}, given above, and $\lambda=\lambda^{LS}$ where λ^{LS} solves

$$\min_{\lambda} \, ||\nabla f - A\lambda||_2^2 \quad .$$

By definition, the SQP step will be a descent direction for L_a if $-s^T\nabla L_a>0$. Note that we use the simplified notation ∇L_a to mean $\nabla_x L_a(x,\lambda,\sigma)$ with λ and σ held constant. Since $s=Ys_Y+Zs_Z$ and $QQ^T=I$, we may write $s^T\nabla L_a$

$$s^T QQ^T\nabla L_a = s^T(YY^T+ZZ^T)\nabla L_a = s_Y{}^T Y^T\nabla L_a + s_Z{}^T Z^T\nabla L_a \quad .$$

Using $A=YR$ and the properties of L_a, Y, and Z yields

$$s^T\nabla L_a = s_Y{}^T Y^T(g-A\lambda) + s_Y{}^T\sigma Rc + s_Z{}^T Z^T g \quad . \tag{2.1}$$

Substituting c, Z^Tg, and λ^{QP} from equations (1.7) into (2.1) and letting $L_a^{QP}=L_a(x,\lambda^{QP},\sigma)$, we have

$$-s^T\nabla L_a^{QP} = s_Y{}^T B_{YY}s_Y + s_Y{}^T B_{YZ}s_Z + s_Z{}^T B_{ZY}s_Y + s_Z{}^T B_{ZZ}s_Z + \sigma s_Y{}^T RR^T s_Y \quad . \tag{2.2}$$

Or, using λ^{LS} and letting $L_a^{LS}=L_a(x,\lambda^{LS},\sigma)$ we get

$$-s^T\nabla L_a^{LS} = s_Z{}^T B_{ZZ}s_Z + s_Z{}^T B_{ZY}s_Y + \sigma s_Y{}^T RR^T s_Y \quad . \tag{2.3}$$

To establish the sign of these quantities we use a result from Debreu (1952).

Lemma 2.1

Define $A \in R^{n \times t}$, $B \in R^{n \times n}$, $z \in R^n$. *Then,* $z^T B z > 0$ *for every* $z \neq 0$ *such that* $A^T z = 0$ *if and only if there exists a number* $\bar{\alpha}$ *such that* $z^T B z + \alpha z^T A A^T z > 0$ *for all* $z \neq 0$ *and* $\alpha > \bar{\alpha}$.

Applying this result, we prove the following theorem.

Theorem 2.2

Suppose s_Y *and* s_Z *are given by (1.7) and* $x \neq x^*$. *If* $s_Z \neq 0$, R *is full rank, and* B_{ZZ} *is positive definite, then there exists a* $\bar{\sigma}$ *such that the SQP step* $s = Y s_Y + Z s_Z$ *is a descent direction for* $L_a(x,\lambda,\sigma)$ *for either* λ^{QP} *or* λ^{LS} *and all* $\sigma > \bar{\sigma}$.

Proof:

We will show $-s^T \nabla L_a^{QP}$ and $-s^T \nabla L_a^{LS}$ given by (2.2) and (2.3) are positive by applying Lemma 2.1. First choose $z^T \doteq (s_Y{}^T, s_Z{}^T)$ and set $A^T = [R^T \, 0^T]$,

$$B = \begin{cases} \begin{bmatrix} B_{YY} & B_{ZY} \\ B_{ZY} & B_{ZZ} \end{bmatrix} & \text{for (2.2)} \\ \begin{bmatrix} 0 & 0 \\ B_{ZY} & B_{ZZ} \end{bmatrix} & \text{for (2.3)} \end{cases}$$

Since B_{ZZ} is positive definite and R has full rank we can assume s_Y and s_Z from (1.7) are well defined and of finite norm. In addition, $A^T z = 0$ for $z \neq 0$ if and only if $s_Y = 0$ since R is of full rank. Thus, for all $s_Z \neq 0$ and $\bar{z}^T \doteq (0^T, s_Z{}^T)$, we have $A^T \bar{z} = 0$ and $\bar{z}^T B \bar{z} = s_Z{}^T B_{ZZ} s_Z > 0$ since B_{ZZ} is positive definite. We conclude from Lemma 2.1 that there exists a $\bar{\sigma}$ such that $-s^T \nabla L_a(x,\lambda,\sigma) > 0$ for λ^{QP} or λ^{LS} and all $\sigma > \bar{\sigma}$. ■

We note that the modified SQP step

$$\begin{aligned} s &= Y s_Y + Z s_Z \\ s_Y &= -R^{-T} c \\ s_Z &= -B_{ZZ}{}^{-1} Z^T g \quad , \end{aligned}$$

used in many of the projected quasi-Newton methods discussed below, may also be shown to be a descent direction for L_a when B_{ZZ} is positive definite.

We may also apply Lemma 2.1 to prove that a quasi-Newton step for the l_2-penalty function, $p \doteq f + \frac{1}{2} \sigma c^T c$ is a descent direction for p if the null space portion of the Hessian approximation is positive definite.

Given this motivation, how do we maintain positivity? In unconstrained quasi-Newton methods, the ability to maintain a positive definite Hessian approximation when far from the solution is a function of the line-search procedure. It can be shown that a sufficiently accurate line search satisfying the Goldstein-Armijo criteria (Goldstein (1967), Armijo (1966)) would guarantee the existence of a positive definite update at each iteration - even far from the solution. In the constrained case, however, if we update using a step in the full space it is nearly impossible to guarantee positivity of the nullspace Hessian approximation by a simple line-search procedure.

To handle this, two basic quasi-Newton approaches for constrained problems have emerged. The earliest methods approximated the full Hessian matrix. Schittkowski (1984) and

Tapia (1978) compared most of the existing algorithms and show that many are equivalent in the equality constrained case. Murray (1969), Biggs (1972, 1975), Garcia Palomares and Mangasarian (1976), Han (1976), Tapia (1977), and Glad (1979) Schittkowski (1981), Mayne and Polak (1982), Powell and Yuan (1986), Gill, Murray, Saunders, and Wright (1985, 1986), and Hestenes (1969) approximated the full Hessian of either the Lagrangian or the augmented Lagrangian. These methods have several drawbacks. Firstly, they require the storage of the full $n \times n$ Hessian matrix though we need at most an $(n-m) \times n$ submatrix to solve for the step from x to x_+. Secondly, whether we update the Hessian of the Lagrangian or the augmented Lagrangian, these methods cannot reliably assure that the Hessian approximation will remain well conditioned and positive definite at each iteration - even near the solution. In part, this may be due to incompatibilities between our chosen merit function and the function whose Hessian we are approximating (Boggs and Tolle, 1987). Powell (1978) overcame this difficulty by introducing a modified BFGS formula which keeps the entire Hessian approximation positive definite. Unfortunately Powell's modification may generate very ill-conditioned updates (Powell, 1985).

Projected methods first suggested by Wright (1976) and Murray and Wright (1978) and more recently by Tanabe (1981), Womersley and Fletcher (1982), Gabay (1982), Coleman and Conn (1984), Nocedal and Overton (1985), and Gurwitz (1986), approximate the projected Hessian, $Z^T \nabla^2 LZ$, and either approximate the quantity $Z^T \nabla^2 LYs_Y$ by finite differences or ignore it. This approach maintains positivity of the null-space approximation without introducing a penalty parameter since we may assume the matrix $Z^T \nabla^2 LZ$ is positive definite near the solution. Thus, we never need to project the full matrix to solve for the step and we solve smaller systems: $(n-m) \times (n-m)$ instead of the $n \times n$ systems which are solved in the full matrix methods.

The methods of Womersley and Fletcher (1982) and Gabay (1982) simply use the projected quantities

$$s = Z^T (x_+ - x)$$
$$y = Z^T (\nabla_x L(x_+) - \nabla_x L(x))$$

to update an approximation to $Z^T \nabla^2 LZ$. Since this choice can not guarantee $s^T y > 0$ they use Powell's modified BFGS update to keep the reduced Hessian approximation positive definite. As noted by Nocedal and Overton (1985), this modification may lead to ill-conditioned updates if $s^T y \gg s^T Bs$ even when B approximates the reduced Hessian.

The projected methods of Coleman and Conn (1984) and Fontecilla (1987) have no penalty parameters or modified updates. They maintain $B_{ZZ} > 0$ by evaluating one extra gradient, $\nabla L(x+h)$, per iteration. Using the projected quantities

$$s = Z^T (x_+ - x) = Z^T (h+v) = Z^T h$$
$$y = Z^T (\nabla L(x+h) - \nabla L(x))$$

and the quasi-Newton equation $B_{ZZ}^+ s_Z = Z^T (\nabla L(x+h) - \nabla L(x))$ they compute either a BFGS or DFP update. This choice insures that $s^T y > 0$ for x near x^* so that B_{ZZ} will remain positive definite near the solution. Unfortunately, when far from the solution, this approach can not guarantee positivity even though two gradients are computed for each iteration.

Nocedal and Overton (1985) avoid extra gradient evaluations by placing special restrictions on the h and v steps. They guarantee positivity of the null space portion of the Hessian, using only projections of the step $s=x_+-x=h+v$ and the corresponding change in the gradient $y=\nabla L(x_+)-\nabla L(x)$ for updating. In their method, updating at iteration k is only allowed when

$$||v_k|| < \frac{\eta}{(k+1)^{1+\nu}} ||h_k||$$

Here, η and ν are arbitrary parameters which must be properly selected for good performance. They prove that this restriction guarantees $s^T y > 0$ in a region very near x^*. However, their result assures frequent updating only very near the solution and may have limited practical computational value.

3. Extending the Variational Approach to Derive New Quasi-Newton Updates for Constrained Optimization

3.1. The Lower-Half Update

Let us now consider approximating the transformed Hessian matrix, (1.6), while maintaining positivity of the null-space approximation, a least change condition, and the transformed quasi-Newton equation $Q^T(B_+)QQ^T s = Q^T y$ or in partitioned form

$$\begin{bmatrix} B_{YY}^+ & B_{YZ}^+ \\ B_{ZY}^+ & B_{ZZ}^+ \end{bmatrix} \begin{bmatrix} s_Y \\ s_Z \end{bmatrix} = \begin{bmatrix} y_Y \\ y_Z \end{bmatrix} \doteq \begin{bmatrix} Y^T(\nabla L(x+s)-\nabla L(x)) \\ Z^T(\nabla L(x+s)-\nabla L(x)) \end{bmatrix} \quad . \tag{3.4}$$

Note that $y \doteq \nabla L(x+s)-\nabla L(x)$ requires gradient information only at x and $x_+=x+s$ - no extra midpoint gradient evaluations are required. Earlier, we noted that the matrices $Y^T\nabla^2 LY$ and $Y^T\nabla^2 LZ$ are required only to solve for the new Lagrangian multipliers - they are not necessary to solve for the Newton step, s. Thus, they can be ignored if we obtain the multipliers in some other way, say by a least squares estimate. With this in mind we first consider estimating only the terms $Z^T\nabla^2 LY$ and $Z^T\nabla^2 LZ$, the lower half of the partitioned form of the Hessian matrix, (1.6). We will approximate $Z^T\nabla^2 LZ$ by a positive definite, symmetric matrix B_{ZZ} and $Z^T\nabla^2 LY$ by a matrix B_{ZY} and require that B_{ZY}^+ and B_{ZZ}^+ satisfy only the lower half of the quasi-Newton equation (3.4)

$$\begin{bmatrix} B_{ZY}^+ & B_{ZZ}^+ \end{bmatrix} \begin{bmatrix} s_Y \\ s_Z \end{bmatrix} = y_Z \quad . \tag{3.5}$$

Our approach is similar to that used in Dennis and Schnabel (1979) for unconstrained updates. Assuming the current approximation, B_{ZZ}, is positive definite and symmetric with a Cholesky decomposition, $B_{ZZ}=LL^T$, we define

$$B_{ZZ}^+ \doteq (L+E)(L+E)^T = J_+J_+^T \tag{3.6}$$

$$B_{ZY}^+ \doteq B_{ZY} + H \tag{3.7}$$

where $H \in R^{(n-t)\times t}$, $E \in R^{(n-t)\times(n-t)}$. Substituting these expressions into the quasi-Newton condition, (3.5), and rearranging, we obtain

$$\begin{bmatrix} H & (L+E)(L+E)^T \end{bmatrix} \begin{bmatrix} s_Y \\ s_Z \end{bmatrix} = y_Z - B_{ZY}s_Y \doteq r_Z \quad ,$$

where $r_Z \in R^{(n-t)}$. Defining $w \doteq (L+E)^T s_Z$, $w \in R^{(n-t)}$ our complete variational formulation is to choose H and E in order to

$$\min_{h,e} \ \tfrac{1}{2} \left|\left| \begin{bmatrix} H & E \end{bmatrix} \right|\right|_F^2$$

$$\text{subject to} \quad \begin{bmatrix} H & (L+E) \end{bmatrix} \begin{bmatrix} s_Y \\ w \end{bmatrix} = r_Z \quad . \tag{3.8}$$

Thus, we have an equality constrained QP in the variables h_{ij} and e_{ij}, the entries of H and E.

Applying the first order optimality conditions to this QP, we determine

$$H = us_Y^T \tag{3.9}$$

and $E = uw^T$ where $u \in R^{(n-t)}$ is the vector of Lagrange multipliers. Substituting E into the definition of w we get, $w = L^T s_Z/(1-s_Z^T u)$ and thus

$$E = \frac{us_Z^T L}{1-s_Z^T u} \quad . \tag{3.10}$$

Substituting H and E from (3.9) and (3.10) into the quasi-Newton equation (3.8), noting $B_{ZZ}=LL^T$ and defining $\alpha \doteq 1/(1-u^T s_Z)$ we may write u as a function of the unknown scalar α,

$$u = \frac{r_Z - \alpha B_{ZZ} s_Z}{s_Y^T s_Y + \alpha^2 s_Z^T B_{ZZ} s_Z} \quad . \tag{3.11}$$

Multiplying by $-s_Z^T$, adding 1 to both sides, and inverting, yields

$$\alpha = \frac{s_Y^T s_Y + \alpha^2 s_Z^T B_{ZZ} s_Z}{s_Y^T s_Y + \alpha^2 s_Z^T B_{ZZ} s_Z - s_Z^T r_Z + \alpha s_Z^T B_{ZZ} s_Z} \quad .$$

Collecting terms in α and noting that $s_Z^T B_{ZZ} s_Z > 0$, we obtain the cubic equation

$$\alpha^3 + \alpha \frac{(s_Y^T s_Y - s_Z^T r_Z)}{s_Z^T B_{ZZ} s_Z} - \frac{s_Y^T s_Y}{s_Z^T B_{ZZ} s_Z} = 0 \quad . \tag{3.12}$$

Thus, given a solution for this cubic in α, we can determine u, H, and E from equations (3.11), (3.9), and (3.10).

In a variational derivation of the standard BFGS formula, we find α is the root of the quadratic equation $\alpha^2-(s^T y/s^T Bs)=0$, making necessary the familiar condition $s^T y>0$ in order to insure a well defined update. In contrast, because equation (3.12) is a cubic, we know it always has at least one real root. Thus, it seems, we are guaranteed that α, u, H, and E will be well defined under all circumstances. The next theorem shows that this generally holds for the Lower-Half Update (LHU). However, as we will discuss below, special circumstances require a more detailed analysis of the cubic root, α.

Theorem 3.1

If $s_Y \neq 0$, $s_Z \neq 0$, and $B_{ZZ}=LL^T$ is positive definite, then the updated matrices B_{ZY}^+ and B_{ZZ}^+, given by equations (3.6), (3.7), (3.9), (3.10), (3.11), and (3.12) are well defined and satisfy the quasi-Newton equation (3.5). In addition, B_{ZZ}^+ is positive definite and symmetric.

Proof:

B_{ZY}^+ and B_{ZZ}^+ are well defined, since a real solution to the cubic equation (3.12) always exists under the stated assumptions.

Symmetry in B_{ZZ}^+ is clear. To see that $B_{ZZ}^+=J_+J_+^T$ is positive definite we must show that $J_+^T=L^T+E^T = L^T(I+\alpha s_Z u^T)$ is nonsingular. Since L is nonsingular, we consider the matrix $I+\alpha s_Z u^T$. By the Sherman-Morrison-Woodbury formula this is nonsingular if and only if $\sigma \neq 0$, where

$$\sigma = 1 + \frac{s_Z{}^T u}{1-s_Z{}^T u} = \frac{1}{1-s_Z{}^T u} \doteq \alpha \quad .$$

But, clearly, $\alpha \neq 0$ since $\bar{\alpha}=0$ is a real root of

$$\alpha^3 + \alpha \frac{(s_Y{}^T s_Y - s_Z{}^T r_Z)}{s_Z{}^T B_{ZZ} s_Z} - \frac{s_Y{}^T s_Y}{s_Z{}^T B_{ZZ} s_Z} = 0 \quad .$$

if and only if $s_Y=0$. Since $s_Y \neq 0$, $s_Z \neq 0$ and B_{ZZ} is positive definite by assumption, the cubic is well defined and $\bar{\alpha} \neq 0$. Thus, we have $\sigma \neq 0$ and J_+^T is nonsingular.

Finally, substituting u from equation (3.11) into expressions (3.9) and (3.10) and using these in (3.8) shows that the quasi-Newton equation is also satisfied. ■

Before we continue with two special cases, let us rewrite the update of B_{ZZ} in a form resembling the standard BFGS update. We can show

$$B_{ZZ}^{+} = B_{ZZ} + \frac{pp^T}{p^T s_Z} - \frac{B_{ZZ} s_Z s_Z{}^T B_{ZZ}{}^T}{s_Z{}^T B_{ZZ} s_Z} \tag{3.13}$$

where

$$\begin{aligned} p &= (1-\theta)\alpha B_{ZZ} s_Z + \theta r_Z \\ p^T s_Z &= \alpha^2 s_Z{}^T B_{ZZ} s_Z \\ \theta &= \frac{\alpha^2 s_Z{}^T B_{ZZ} s_Z}{s_Y{}^T s_Y + \alpha^2 s_Z{}^T B_{ZZ} s_Z} \quad . \end{aligned} \tag{3.14}$$

We use this formulation in the proof of the following corollary which shows that for the special case $s_Y=0$, the LHU is naturally equivalent to a projected BFGS update of the matrix B_{ZZ} which is well defined if $s_Z{}^T y_Z > 0$.

Corollary 3.2

Assume we now have $s_Y=0$ and $s_Z{}^T r_Z = s_Z{}^T y_Z > 0$ which implies $s_Z \neq 0$. As before, assume $B_{ZZ}=LL^T$ is positive definite. Then the updated matrices B_{ZY}^{+} and B_{ZZ}^{+}, given by equations (3.6), (3.7), (3.9), (3.10), (3.11), and (3.12) are well defined and satisfy the quasi-Newton equation (3.5). In addition, $B_{ZY}^{+}=B_{ZY}$, B_{ZZ}^{+} is given by the projected BFGS update

$$B_{ZZ}^{+} = B_{ZZ} + \frac{y_Z y_Z{}^T}{s_Z{}^T y_Z} - \frac{B_{ZZ} s_Z s_Z{}^T B_{ZZ}{}^T}{s_Z{}^T B_{ZZ} s_Z} \quad . \tag{3.15}$$

which is symmetric and positive definite.

Proof:

Symmetry is obvious and, since $s_Y=0$, the quasi-Newton condition reduces to $B_{ZZ}^{+} s_Z = y_Z$ which is clearly satisfied by equation (3.15).

Since $s_Y=0$, equation (3.9) gives $H=0$, so H is well defined. Considering equation (3.12), we see that for $s_Y=0$ the cubic degenerates to the "quadratic" equation

$$\alpha^3 - \alpha \frac{s_Z{}^T r_Z}{s_Z{}^T B_{ZZ} s_Z} = 0 \quad .$$

Using the definition of r_Z and simplifying we get the familiar expression

$$\alpha = \sqrt{s_Z{}^T y_Z / s_Z{}^T B_{ZZ} s_Z} \quad .$$

which is well defined when $s_Z{}^T y_Z > 0$. Substituting this value for α into (3.14), setting

s_Y=0, and using the definition of r_Z, we have $p^T s_Z = s_Z{}^T y_Z$, θ=1, and $p = y_Z$. Then (3.13) gives the projected BFGS formula

$$B_{ZZ}^{+} = B_{ZZ} + \frac{y_Z y_Z{}^T}{s_Z{}^T y_Z} - \frac{B_{ZZ} s_Z s_Z{}^T B_{ZZ}{}^T}{s_Z{}^T B_{ZZ} s_Z} \quad .$$

Thus, when s_Y=0, B_{ZZ}^{+} is positive definite if and only if $s_Z{}^T y_Z > 0$. ■

The next Corollary shows that new information may be incorporated into the Hessian approximation even when s_Z=0.

Corollary 3.3

If s_Z=0 and $s_Y \neq 0$ then $B_{ZZ}^{+} = B_{ZZ}$. Furthermore, B_{ZY}^{+} given by the projected Broyden update

$$B_{ZY}^{+} = B_{ZY} + \frac{(y_Z - B_{ZY} s_Y) s_Y{}^T}{s_Y{}^T s_Y} \quad , \tag{3.16}$$

is well defined and satisfies the quasi-Newton equation.

Proof:

From (3.10) we immediately see E=0 when s_Z=0 so that $B_{ZZ}^{+} = B_{ZZ}$.

Equation (3.11) simplifies to

$$u = \frac{r_Z}{s_Y{}^T s_Y} = \frac{(y_Z - B_{ZY} s_Y)}{s_Y{}^T s_Y}$$

giving

$$B_{ZY}^{+} = B_{ZY} + \frac{(y_Z - B_{ZY} s_Y) s_Y{}^T}{s_Y{}^T s_Y}$$

which is well defined for all $s_Y \neq 0$. From (3.16) we see the simplified quasi-Newton equation, $B_{ZY}^{+} s_Y = y_Z$, is satisfied. ■

For some algorithms, (e.g. transformation methods based on the l_2-penalty function), we may wish to approximate all four projections of the Hessian matrix. Although the LHU approximates only two blocks of the projected Hessian, we can use the LHU in a two-stage updating procedure to approximate $Y^T \nabla^2 L Y$ and $Y^T \nabla^2 L Z$ by the matrices B_{YY} and B_{YZ}. Suppose we have updated B_{ZZ} to B_{ZZ}^{+} and B_{ZY} to B_{ZY}^{+} using the LHU. We want our complete update to satisfy the full projected quasi-Newton equation (3.4). Since the lower half is already satisfied by the LHU we concern ourselves with satisfying

$$\begin{bmatrix} B_{YY}^{+} & B_{YZ}^{+} \end{bmatrix} \begin{bmatrix} s_Y \\ s_Z \end{bmatrix} = y_Y \quad .$$

For symmetry we take $B_{YY}^{+} = B_{YY} + (G + G^T)$ and $B_{YZ}^{+} = B_{ZY}^{+}{}^T$ where B_{ZY}^{+} has been defined by the LHU. We then rewrite the quasi-Newton equation and solve the QP

$$\min_G \; \| G + G^T \|_F^2$$

$$\text{subject to} \quad (G + G^T) s_Y = y_Y - B_{YY} s_Y - B_{YZ}^{+} s_Z \doteq r_Y \quad .$$

The solution, given by the projected PSB update

$$G+G^T = \frac{r_Y s_Y{}^T + s_Y r_Y{}^T}{s_Y{}^T s_Y} - \frac{s_Y{}^T r_Y \, s_Y s_Y{}^T}{(s_Y{}^T s_Y)^2} \quad ,$$

allows us to define an update of the complete projected Hessian satisfying the full quasi-Newton equation, (3.4).

3.2. The Partitioned BFGS Update

Returning to the partitioned form of the nonlinear equality problem, we derive another formula which simultaneously updates all four projections of the Hessian matrix.

As we did in the Lower-Half Update, we will approximate $Z^T\nabla^2 LZ$ by a positive definite, symmetric matrix B_{ZZ}. To maintain symmetry in our approximation we require that B_{YY} remain symmetric and force symmetry in the off-diagonal blocks by setting $B_{YZ}=B_{ZY}{}^T$. We also require that B_{YY}^+, B_{YZ}^+, B_{ZY}^+, and B_{ZZ}^+ satisfy the projected quasi-Newton equation, (3.4). Again, assume the current approximation $B_{ZZ}=LL^T$ is positive definite and symmetric and then define

$$B_{ZZ}^+ = (L+E)(L+E)^T = J_+J_+{}^T \tag{3.17}$$

$$B_{ZY}^+ = B_{ZY} + H = B_{YZ}^+{}^T \tag{3.18}$$

$$B_{YY}^+ = B_{YY} + (G+G^T) \quad . \tag{3.19}$$

Defining $w \doteq (L+E)^T s_Z$, $w \in R^{(n-t)}$ and assuming $L+E$ nonsingular, the quasi-Newton condition, (3.4), may be written

$$\begin{bmatrix} (G+G^T) & H^T & 0 \\ H & 0 & (L+E) \end{bmatrix} \begin{bmatrix} s_Y \\ s_Z \\ w \end{bmatrix} = \begin{bmatrix} y_Y - B_{YY}s_Y - B_{YZ}s_Z \\ y_Z - B_{ZY}s_Y \end{bmatrix} \doteq \begin{bmatrix} r_Y \\ r_Z \end{bmatrix} \quad . \tag{3.20}$$

where $r_Y \in R^t$ and $r_Z \in R^{(n-t)}$. The variational formulation is completed by choosing G, H, and E to

$$\min_{g,h,e} \ 1/4 \left\| \begin{bmatrix} (G+G^T) & H^T \\ H & E \end{bmatrix} \right\|_F^2$$

subject to the quasi-Newton condition (3.20).

Forming the Lagrangian in the variables g_{ij}, h_{ij} and e_{ij} and two Lagrange multipliers $u_Y \in R^t$ and $u_Z \in R^{(n-t)}$, differentiating, and applying the first order conditions, we obtain

$$G+G^T = u_Y s_Y{}^T + s_Y u_Y{}^T \tag{3.21}$$

$$H = u_Z s_Y{}^T + s_Z u_Y{}^T \tag{3.22}$$

$$E = 2u_Z w^T \quad .$$

Using E in the definition of w we get $w = L^T s_Z/(1-2s_Z{}^T u_Z)$ and

$$E = \frac{2u_Z s_Z{}^T L}{1-2s_Z{}^T u_Z} \quad . \tag{3.23}$$

Substituting H, $G+G^T$, and E from (3.21), (3.22), and (3.23) into the quasi-Newton equation (3.20) yields

$$u_Y s_Y{}^T s_Y + s_Y u_Y{}^T s_Y + s_Y u_Z{}^T s_Z + u_Y s_Z{}^T s_Z = r_Y$$

$$u_Z s_Y{}^T s_Y + s_Z u_Y{}^T s_Y + \alpha B_{ZZ} s_Z + 2\alpha^2 u_Z s_Z{}^T B_{ZZ} s_Z = r_Z$$

where $B_{ZZ}=LL^T$ and $\alpha \doteq 1/(1-2s_Z{}^T u_Z)$. Solving these equations for u_Y and u_Z in terms of the unknown scalars $u_Z{}^T s_Z$ and $u_Y{}^T s_Y$ gives,

$$u_Y = \frac{r_Y - s_Y(u_Y{}^T s_Y + u_Z{}^T s_Z)}{s^T s} \tag{3.24}$$

$$u_Z = \frac{r_Z - s_Z u_Y{}^T s_Y - \alpha B_{ZZ} s_Z}{s_Y{}^T s_Y + 2\alpha^2 s_Z{}^T B_{ZZ} s_Z} \tag{3.25}$$

where $s^T s \doteq s_Y{}^T s_Y + s_Z{}^T s_Z$. Multiplying (3.24) by $s_Y{}^T$ we solve for $u_Y{}^T s_Y \doteq \gamma$ as a function of $u_Z{}^T s_Z$.

$$\gamma \doteq u_Y{}^T s_Y = \frac{s_Y{}^T r_Y - s_Y{}^T s_Y u_Z{}^T s_Z}{s_Y{}^T s_Y + s^T s} \quad . \tag{3.26}$$

Substituting γ into (3.25), multiplying both sides by $-2s_Z{}^T$, adding 1 to both sides and simplifying, gives us the cubic

$$\alpha^3 + \alpha\eta - \nu = 0 \tag{3.27}$$

where $\alpha = 1/(1-2s_Z{}^T u_Z)$ and

$$\eta \doteq \frac{s_Y{}^T s_Y (s_Y{}^T s_Y - 2s_Z{}^T r_Z) + s_Z{}^T s_Z (s_Y{}^T r_Y - s_Z{}^T r_Z)}{s_Z{}^T B_{ZZ} s_Z (s_Y{}^T s_Y + s^T s)}$$

$$\nu \doteq \frac{(s_Y{}^T s_Y)^2}{s_Z{}^T B_{ZZ} s_Z (s_Y{}^T s_Y + s^T s)}$$

Given a solution for this cubic, we can compute $u_Z{}^T s_Z$ from α and then γ, u_Y, u_Z, $G+G^T$, H, and E from equations (3.26), (3.24), (3.25), (3.21), (3.22), and (3.23).

As we found for the LHU, this cubiç will have at least one real root when $s_Z \neq 0$. Carrying out an analysis similar to that for the LHU yields an analog of Theorem 3.1 showing the Partitioned BFGS Update (PBU) is well defined when $s_Y \neq 0$ and $s_Z \neq 0$. Since the proofs are nearly identical to those for the LHU, they are omitted from the following Theorem and Corollaries.

Theorem 3.4

If $s_Y \neq 0$, $s_Z \neq 0$, and $B_{ZZ} = LL^T$ is positive definite, then the updated matrices B_{YY}^+, B_{YZ}^+, B_{ZY}^+, and B_{ZZ}^+, given by equations (3.17), (3.18), (3.19), (3.21), (3.22), (3.23), (3.24), (3.25), (3.26), and (3.27) are well defined and satisfy the quasi-Newton equation (3.20). In addition, B_{ZZ}^+ is positive definite and symmetric.

Analogs of Corollaries 3.2 and 3.3 show the PBU is naturally equivalent to projected versions of standard updates when either $s_Y = 0$ or $s_Z = 0$.

Corollary 3.5

Assume now $s_Y = 0$ and $s_Z{}^T r_Z = s_Z{}^T y_Z > 0$ which implies $s_Z \neq 0$. As before, also assume $B_{ZZ} = LL^T$ is positive definite. Then the updated matrices B_{YY}^+, B_{YZ}^+, B_{ZY}^+, and B_{ZZ}^+, given by equations (3.17), (3.18), (3.19), (3.21), (3.22), (3.23), (3.24), (3.25), (3.26), and (3.27) are well defined and satisfy the full quasi-Newton equation (3.4). In addition, $B_{YY}^+ = B_{YY}$, and $B_{YZ}^+ = B_{ZY}^{+T}$ is given by the projected Broyden

$$B_{YZ}^+ = B_{YZ} + \frac{(y_Y - B_{YZ} s_Z) s_Z{}^T}{s_Z{}^T s_Z} \quad .$$

Furthermore, B_{ZZ}^+ given by the projected BFGS formula

$$B_{ZZ}^+ = B_{ZZ} + \frac{y_Z y_Z{}^T}{s_Z{}^T y_Z} - \frac{B_{ZZ} s_Z s_Z{}^T B_{ZZ}{}^T}{s_Z{}^T B_{ZZ} s_Z} \quad ,$$

is symmetric and positive definite.

Corollary 3.6

If $s_Z=0$ and $s_Y\neq 0$ then the updated matrices are well defined and satisfy the full quasi-Newton equation. In addition, $B_{ZZ}^+=B_{ZZ}$, and $B_{ZY}^+=B_{YZ}^{+T}$ is given by the projected Broyden formula,

$$B_{ZY}^+ = B_{ZY} + \frac{(y_Z - B_{ZY}s_Y)s_Y^T}{s_Y^T s_Y} \quad ,$$

while B_{YY} is given by the projected PSB,

$$B_{YY}^+ = B_{YY} + \frac{(y_Y - B_{YY}s_Y)s_Y^T + s_Y(y_Y - B_{YY}s_Y)^T}{s_Y^T s_Y} - \frac{s_Y^T(y_Y - B_{YY}s_Y)\, s_Y s_Y^T}{(s_Y^T s_Y)^2} \quad .$$

Again it is useful to write B_{ZZ}^+ in the standard form

$$B_{ZZ}^+ = B_{ZZ} + \frac{pp^T}{p^T s_Z} - \frac{B_{ZZ}s_Z s_Z^T B_{ZZ}^T}{s_Z^T B_{ZZ} s_Z}$$

where

$$p = (1-\theta)\alpha B_{ZZ}s_Z + \theta(r_Z - \gamma s_Z)$$
$$p^T s_Z = \alpha^2 s_Z^T B_{ZZ} s_Z$$

where α is given by (3.27) and

$$\theta = \frac{2\alpha^2 s_Z^T B_{ZZ} s_Z}{s_Y^T s_Y + 2\alpha^2 s_Z^T B_{ZZ} s_Z} \quad .$$

In the next section we will present an estimate of the condition number of B_{ZZ} which may be computed before actually updating. This permits us skip either the LHU or PBU update when the condition number estimate is too high. When using the PBU formula another alternative is available: update only the blocks B_{YY}, B_{YZ}, and B_{ZY} in a way which maintains the quasi-Newton condition for the upper half of the partitioned matrix without affecting the condition number of B_{ZZ}. Consider a modified formulation of the PBU

$$\min_{g,h,e} \ \tfrac{1}{4} \left\| \begin{bmatrix} (G+G^T) & H^T \\ H & 0 \end{bmatrix} \right\|_F^2$$

subject to the quasi-Newton condition

$$[(G+G^T) \ \ H^T] \begin{bmatrix} s_Y \\ s_Z \end{bmatrix} = y_Y - B_{YY}s_Y - B_{YZ}s_Z \doteq r_Y \quad .$$

Introducing the Lagrange multiplier vector u_Y and carrying through the derivation as we did earlier, we obtain the following Modified Partitioned BFGS Update (MPBU):

$$\begin{aligned} B_{YY}^+ &= B_{YY} + (G+G^T) = B_{YY} + u_Y s_Y^T + s_Y u_Y^T \\ B_{YZ}^+ &= B_{YZ} + H^T = B_{YZ} + u_Y s_Z^T = B_{ZY}^{+T} \end{aligned} \qquad \text{(MPBU)}$$

where

$$u_Y = \frac{r_Y - s_Y\gamma}{s_Y^T s_Y + s_Z^T s_Z}$$
$$\gamma = \frac{s_Y^T r_Y}{s_Y^T s_Y + s^T s} \quad .$$

In those cases where the standard PBU cannot be performed, this update incorporates new information about B_{YY} and B_{YZ} without creating any ill-conditioning.

4. Analysis of the Cubic Equations in the LHU and PBU

In this section we present the detailed analysis required to implement the new update formulas for the LHU and PBU. We need to consider two details: choosing a root for the cubic when more than one real root is available and avoiding ill-conditioning when the step $v{=}Ys_Y{\cong}0$ while $s_Z{}^T y_Z{<}0$. We will see in this chapter that these problems are complementary: $s_Y{\cong}0$ with $s_Z{}^T y_Z{<}0$ only occurs when a single real root exists for the cubic equation.

We begin by studying the cubic equation in full detail to determine when multiple real roots are possible and how those roots behave as we approach the two special cases $s_Y{=}0$ and $s_Z{=}0$. Actually we will consider these special cases more generally by considering the cubic equation when either $||s_Y|| << ||s_Z||$ or $||s_Z|| << ||s_Y||$. Next, we will consider root selection when three real roots are available and finally discuss the rejection of a single real root which might cause ill-conditioning in our Hessian approximation.

4.1. Solving the Cubic Equation

The roots of polynomial equations with real coefficients may be determined iteratively. For a general discussion see Brent (1973), Householder (1970), Press, et al. (1986), and Ralston and Rabinowitz (1978).

The cubic equations (3.12) and (3.27) must be solved to compute either the LHU or PBU update. In the standard form,

$$c(\alpha) \doteq \alpha^3 + \pi\alpha^2 + \eta\alpha + \nu \quad ,$$

we have for the PBU:

$$\begin{aligned} \pi &= 0 \\ \eta &= \frac{s_Y{}^T s_Y (s_Y{}^T s_Y - 2 s_Z{}^T r_Z) + s_Z{}^T s_Z (s_Y{}^T r_Y - s_Z{}^T r_Z)}{s_Z{}^T B_{ZZ} s_Z (s_Y{}^T s_Y + s^T s)} \\ \nu &= -\frac{(s_Y{}^T s_Y)^2}{s_Z{}^T B_{ZZ} s_Z (s_Y{}^T s_Y + s^T s)} \end{aligned} \tag{c-PBU}$$

and for the LHU:

$$\begin{aligned} \pi &= 0 \\ \eta &= \frac{(s_Y{}^T s_Y - s_Z{}^T r_Z)}{s_Z{}^T B_{ZZ} s_Z} \\ \nu &= -\frac{s_Y{}^T s_Y}{s_Z{}^T B_{ZZ} s_Z} \end{aligned} \tag{c-LHU}$$

with $s^T s \doteq s_Y{}^T s_Y + s_Z{}^T s_Z$.

We immediately note that these equations have a rather special form. First, since the ν-terms in both cubics involve only positive quantities, we know that ν will always be negative. This will play an important role as we analyze the roots in the special cases s_Y or s_Z near zero. Secondly, since $\pi{=}0$ for both cubics, the analytical solution for the three roots is greatly simplified. Using the solution in Abramowitz and Stegun (1964) we obtain,

$$A, B = [-\tfrac{1}{2}\nu \pm (\omega)^{1/2}]^{1/3} \tag{4.1}$$

with

$$\omega = \frac{1}{27}\eta^3 + \frac{1}{4}\nu^2 \tag{4.2}$$

which yields the roots:

$$\begin{aligned} \alpha_1 &= A + B \\ \alpha_2 &= -\frac{1}{2}(A+B) + \frac{\sqrt{-3}}{2}(A-B) \\ \alpha_3 &= -\frac{1}{2}(A+B) - \frac{\sqrt{-3}}{2}(A-B) \quad . \end{aligned} \tag{4.3}$$

The number of real roots can be determined from the sign of ω as follows:

$\omega > 0$, => 1 real root, 2 complex conjugate roots

$\omega = 0$, => 3 real roots, at least 2 equal

$\omega < 0$, => 3 real distinct roots .

Note that the cubic always has at least one real root. In Fenyes (1987) we also used the relations

$$\begin{aligned} \alpha_1+\alpha_2+\alpha_3 &= -\pi \\ \alpha_1\alpha_2+\alpha_1\alpha_3+\alpha_3\alpha_2 &= \eta \\ \alpha_1\alpha_2\alpha_3 &= -\nu \end{aligned} \tag{4.4}$$

to obtain safeguarded formulas for accurately evaluating these roots in the general case.

4.1.1. Special Cases: $s_Y << s_Z$ or $s_Z << s_Y$

As s_Y or s_Z approach zero the cubic equations and the new updates require some additional analysis. When $s_Y \cong 0$ or $s_Y << s_Z$ we take a step mainly in the null space as if we were solving an unconstrained problem in the null space. Since we have obtained little information about the Hessian in the range space, we will want B_{YY} to remain virtually unchanged so that only the null-space partitions, B_{YZ}, B_{ZY}, and B_{ZZ}, should be modified. If $s_Y \cong 0$ and $s_Z^T y_Z > 0$ we will want our new formulas for B_{ZZ} to reduce to a projected form of the standard BFGS formula. Similarly, when $s_Z \cong 0$ or $s_Z << s_Y$ we have gained very little new information in the null space so that our new formulas should leave B_{ZZ} unchanged and update only the partitions of the Hessian containing range-space information: B_{YY}, B_{YZ}, and B_{ZY}. The goal of this section is to insure that the chosen cubic root is continuous and corresponds to a reasonable update in both of these cases.

- $s_Y << s_Z$

First we will study the roots of the cubic when s_Y is zero or nearly zero or $s_Y << s_Z$. Assuming $s_Z \neq 0$, for both the LHU and PBU we see that $\nu \to 0$ as $s_Y \to 0$. It is also clear that η is well behaved for both updates. In fact we see that as $s_Y \to 0$, $\eta \to -s_Z^T r_Z / s_Z^T B_{ZZ} s_Z$ for both the LHU and PBU. We examine the behavior of the cubic roots when s_Y is small by expanding A and B in Taylor series about ν=0. These Taylor expansions were obtained using MACSYMA (Bogen, 1983) a symbolic manipulation program developed at the MIT Laboratory for Computer Science. We consider three cases: $\omega>0$, $\omega=0$, and $\omega<0$.

  - $s_Y << s_Z$, $\omega>0$

Let us first consider $\omega>0$. Since $\omega>0$, A and B are both real. Their Taylor series expansions are

$$A = \frac{1}{\sqrt{3}}\eta^{1/2} - \frac{1}{2}\frac{\nu}{\eta} + \frac{\sqrt{3}}{8}\frac{\nu^2}{\eta^{5/2}} + \frac{1}{2}\frac{\nu^3}{\eta^4} + \ldots$$

$$B = -\frac{1}{\sqrt{3}}\eta^{1/2} - \frac{1}{2}\frac{\nu}{\eta} - \frac{\sqrt{3}}{8}\frac{\nu^2}{\eta^{5/2}} + \frac{1}{2}\frac{\nu^3}{\eta^4} + \ldots \quad .$$

Computing $A+B$, $A-B$ and substituting in (4.3) gives the three roots

$$\alpha_1 = -\frac{\nu}{\eta} + \frac{\nu^3}{\eta^4} + \ldots$$

$$\alpha_2, \alpha_3 = -\frac{1}{2}\left(-\frac{\nu}{\eta} + \frac{\nu^3}{\eta^4} + \ldots\right) \pm i\left(\eta^{1/2} + \frac{3}{8}\frac{\nu^2}{\eta^{5/2}} + \ldots\right) \quad .$$

Clearly, as ν approachs zero, the single real root,

$$\alpha_1 \to 0$$

along the real axis. Since $\eta>0$ when $\omega>0$ and ν is small, α_2 and α_3 approach the pure imaginary roots,

$$\alpha_2, \alpha_3 \to \pm i\, \eta^{1/2}$$

as $\nu \to 0$. However, $\eta \to -s_Z{}^T y_Z / s_Z{}^T B_{ZZ} s_Z$ as $\nu \to 0$ and since ν is small, $\omega>0$ implies both $\eta>0$ and $s_Z{}^T y_Z<0$ which yields

$$\alpha_2, \alpha_3 \to \pm i\, \eta^{1/2} \to \pm i \left[\frac{-s_Z{}^T y_Z}{s_Z{}^T B_{ZZ} s_Z}\right]^{1/2} = \pm i \left[\frac{|s_Z{}^T y_Z|}{s_Z{}^T B_{ZZ} s_Z}\right]^{1/2} \quad .$$

Considering the cubics c-PBU and c-LHU, we see that when $\nu = s_Y \equiv 0$ the cubic degenerates to a quadratic with roots corresponding to α_2 and α_3. The first cubic root, $\alpha_1=0$, is not a meaningful root. In fact our derivations of the cubics in the previous section are not valid for $s_Y=0$ and $\alpha=0$, since the solution for the multiplier vector u requires division by $s_Y{}^T s_Y + \alpha^2 s_Z{}^T B_{ZZ} s_Z \equiv 0$. Interestingly, the two remaining roots, α_2 and α_3, are related to the standard projected BFGS roots, α_{BFGS}. To see this, we note that the variational derivation of the standard projected BFGS update requires the solution of the quadratic

$$\alpha_{BFGS}^2 = s_Z{}^T y_Z / s_Z{}^T B_{ZZ} s_Z \quad .$$

The roots of this quadratic are real and define a positive definite BFGS update if and only if $s_Z{}^T y_Z>0$. When $s_Z{}^T y_Z<0$ we obtain the roots

$$\alpha_{BFGS} = \pm i \left[\frac{|s_Z{}^T y_Z|}{s_Z{}^T B_{ZZ} s_Z}\right]^{1/2}$$

which are identical to the roots α_2 and α_3 obtained when $\nu \equiv 0$.

In Figure 4.1 we have illustrated the roots α_1, α_2 and α_3 for $\nu \to 0$. As $\nu \to 0$, α_2 and α_3 approach the pure imaginary roots α_{BFGS}. Since they are imaginary, we can not use them to define our update. If $\nu \neq 0$, the real root α_1 will be very small but nonzero. Theoretically it is a valid root to the cubic equation. However, as we will show in Section 2.2, using this root may lead to severe ill-conditioning in the matrix B_{ZZ}. As a result, we may not be able to perform the LHU or PBU updates when this situation occurs.

- $s_Y \ll s_Z$, $\omega=0$

The case $\omega=0$ as $\nu \to 0$ is relatively simple. If $\omega \equiv 0$, then $A=B=-(\nu/2)^{1/3}$ which clearly goes to zero as ν does. In this case (4.3) yields three real roots, $\alpha_1, \alpha_2, \alpha_3 \to 0$. As in the previous case, α_1 is not a meaningful root and as $\alpha_2, \alpha_3 \to 0$, ill-conditioning of B_{ZZ} may occur. This situation corresponds to a failure of the standard projected BFGS update which occurs when $s_Z{}^T y_Z \to 0$. To see this we note that in the LHU and PBU formulas imply that

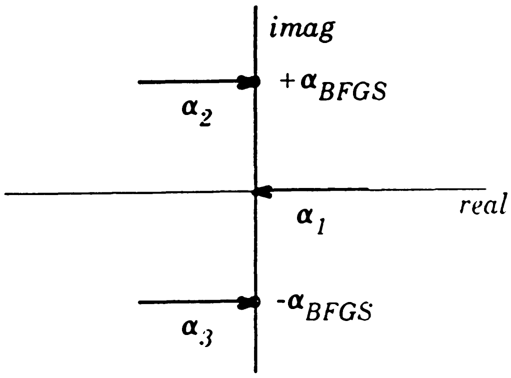

FIGURE 4.1 - CUBIC ROOTS: $\omega>0$ AS $\nu\to 0$

$\eta\to -s_Z^T y_Z / s_Z^T B_{ZZ} s_Z \to 0$ as $\nu\to 0$ since we are considering $\omega\equiv 0$.

- $s_Y << s_Z$, $\omega<0$

Finally, we consider the case that $\nu\to 0$ and $\omega<0$. In this case, A and B are complex conjugates. Thus, the roots simplify to

$$\begin{aligned} \alpha_1 &= 2\,real(A) \\ \alpha_2 &= -real(A) - \sqrt{3}\,imag(A) \\ \alpha_3 &= -real(A) + \sqrt{3}\,imag(A) \end{aligned} \tag{4.5}$$

Again, we examine the behavior of the roots by looking at a Taylor series expansion of A. For $\omega<0$ things are complicated by the fact that $\sqrt{\omega}$ is imaginary, however, we can perform the needed Taylor expansions by considering the real and imaginary parts of A separately. We have,

$$\begin{aligned} real(A) &= \frac{1}{2}(-\eta)^{1/2} + \frac{1}{4}\frac{\nu}{\eta} - \frac{3}{16}\frac{\nu}{(-\eta)^{5/2}} - \frac{1}{4}\frac{\nu^3}{\eta^4} + \ldots \\ imag(A) &= \frac{1}{2\sqrt{3}}(-\eta)^{1/2} - \frac{\sqrt{3}}{4}\frac{\nu}{\eta} - \frac{\sqrt{3}}{16}\frac{\nu}{(-\eta)^{5/2}} + \frac{\sqrt{3}}{4}\frac{\nu^3}{\eta^4} + \ldots \quad . \end{aligned}$$

Of course we also have $real(B)=real(A)$ and $imag(B)=-imag(A)$ since A and B are complex conjugates. From equations (4.5) we obtain the three real roots

$$\begin{aligned} \alpha_1 &= (-\eta)^{1/2} + \frac{1}{2}\frac{\nu}{\eta} - \frac{3}{8}\frac{\nu}{(-\eta)^{5/2}} - \frac{1}{2}\frac{\nu^3}{\eta^4} + \ldots \\ \alpha_2 &= -(-\eta)^{1/2} + \frac{1}{2}\frac{\nu}{\eta} + \frac{3}{8}\frac{\nu}{(-\eta)^{5/2}} - \frac{1}{2}\frac{\nu^3}{\eta^4} + \ldots \\ \alpha_3 &= -\frac{\nu}{\eta} + \frac{\nu^3}{\eta^4} + \ldots \end{aligned} \tag{4.6}$$

Analyzing these roots, it is clear that as $\nu\to 0$, $\alpha_3\to 0$ and $\alpha_1, \alpha_2 \to \pm(-\eta)^{1/2}$. Noting

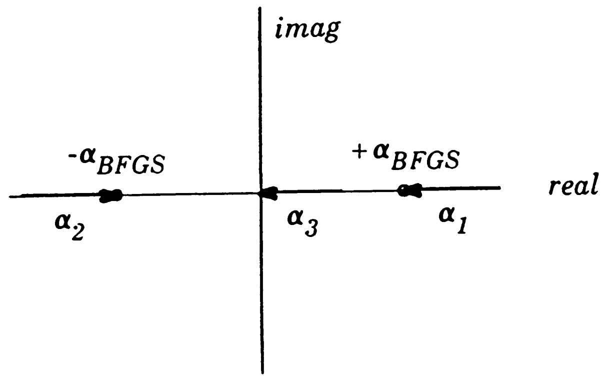

FIGURE 4.2 - CUBIC ROOTS: $\omega<0$ AS $\nu\to 0$

$-\eta\to s_Z{}^T y_Z / s_Z{}^T B_{ZZ} s_Z$ as $\nu\to 0$, we conclude $\alpha_1,\alpha_2\to\alpha_{BFGS}$ where the α_{BFGS} are real roots given by

$$\alpha_{BFGS} = \left[\frac{|s_Z{}^T y_Z|}{s_Z{}^T B_{ZZ} s_Z}\right]^{1/2}$$

since $\omega<0$ implies $s_Z{}^T y_Z>0$. These roots are illustrated in Figure 4.2. In this case, choosing to update with α_1 or α_2 guarantees that as $\nu\to 0$ the PBU and LHU will reduce to the standard projected BFGS update on B_{ZZ}, a projected Broyden on B_{ZY}, and $B_{YY}^{+}\to B_{YY}$.

- $s_Z \ll s_Y$

Next we consider the cubic when s_Z is either zero or nearly zero. We can see that the cubic $c(\alpha)$ defined above blows up as $s_Z\to 0$ so that it must be analyzed in a slightly different form. Assuming $\alpha\neq 0$ we let $\zeta=1/\alpha$. Recalling that $\pi=0$ for both the LHU and the PBU we can make a change of variables in $c(\alpha)$ and solve for the three roots of

$$c(\zeta) = \zeta^3+\frac{\eta}{\nu}\zeta^2+\frac{1}{\nu} = 0 \quad .$$

Or letting $\bar{\nu}\doteq 1/\nu$ and $\bar{\eta}\doteq \eta/\nu$ we have

$$c(\zeta) = \zeta^3+\bar{\eta}\zeta^2+\bar{\nu} = 0 \quad .$$

Given the roots ζ_i, we can set $\alpha_i = 1/\zeta_i$. Using the same analysis we used for the case $s_Y\to 0$ we find that as $\bar{\nu}\to 0$ or $s_Z\to 0$ we have the three roots

$$\zeta_1 = -\bar{\eta} - \frac{\bar{v}}{\bar{\eta}^2} + 2\frac{\bar{v}^2}{\bar{\eta}^5} - 7\frac{\bar{v}^3}{\bar{\eta}^8} + \ldots$$
$$\zeta_2 = \frac{1}{2}\frac{\bar{v}}{\bar{\eta}^2} - \frac{\bar{v}^2}{\bar{\eta}^5} + i\left(\frac{\bar{v}}{\bar{\eta}}\right)^{1/2}\left(1 - \frac{5}{8}\frac{\bar{v}}{\bar{\eta}^3} + \frac{231}{128}\frac{\bar{v}^2}{\bar{\eta}^6}\right) + \ldots$$
$$\zeta_3 = \frac{1}{2}\frac{\bar{v}}{\bar{\eta}^2} - \frac{\bar{v}^2}{\bar{\eta}^5} - i\left(\frac{\bar{v}}{\bar{\eta}}\right)^{1/2}\left(1 - \frac{5}{8}\frac{\bar{v}}{\bar{\eta}^3} + \frac{231}{128}\frac{\bar{v}^2}{\bar{\eta}^6}\right) + \ldots$$

As $s_Z \to 0$ we have $\bar{v} \to 0$ and $\bar{\eta} \to -1$ so that $\zeta_1 \to 1$ and $\zeta_2, \zeta_3 \to 0 \pm i0$. These roots are shown in Figure 4.3.

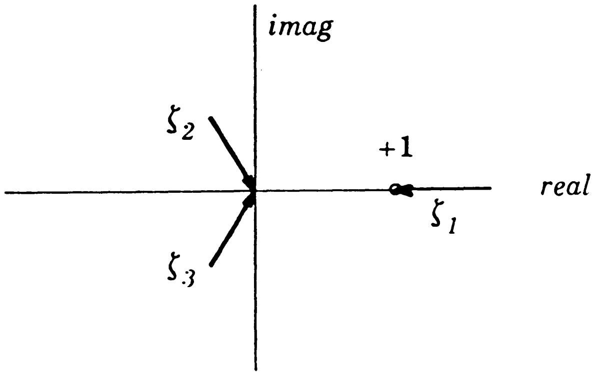

FIGURE 4.3 - CUBIC ROOTS: $s_Z \to 0$

Thus, $\alpha_1 \to 1$ and $\alpha_2, \alpha_3 \to \infty \pm i\infty$. Since α_2 and α_3 are complex for all $\bar{v}$ they are not acceptable roots. Let us examine the real root $\alpha_1 = 1$ in more detail. Recall the standard form

$$B_{ZZ}^{+} = B_{ZZ} + \frac{pp^T}{p^T s_Z} - \frac{B_{ZZ} s_Z s_Z^T B_{ZZ}^T}{s_Z^T B_{ZZ} s_Z} \tag{4.7a}$$

where

$$\begin{aligned} p &= (1-\theta)\alpha B_{ZZ} s_Z + \theta(r_Z - \gamma s_Z) \\ p^T s_Z &= \alpha^2 s_Z^T B_{ZZ} s_Z \end{aligned} \tag{4.7b}$$

and

$$\begin{aligned} \theta_{LHU} &= \frac{\alpha^2 s_Z^T B_{ZZ} s_Z}{s_Y^T s_Y + \alpha^2 s_Z^T B_{ZZ} s_Z} \\ \theta_{PBU} &= \frac{2\alpha^2 s_Z^T B_{ZZ} s_Z}{s_Y^T s_Y + 2\alpha^2 s_Z^T B_{ZZ} s_Z} \quad . \end{aligned} \tag{4.7c}$$

When $s_Z \equiv 0$, $\alpha \equiv 1$ and we have $\theta = 0$ and $p = B_{ZZ} s_Z$ which yields $B_{ZZ}^{+} \equiv B_{ZZ}$ as desired. Also, as $s_Z \to 0$ we see that $\theta \to 0$ and $\alpha_1 \to 1$ for both the LHU and PBU. The equations above then show $p \to B_{ZZ} s_Z$ and $p^T s_Z \to s_Z^T B_{ZZ} s_Z$ so that $B_{ZZ}^{+} \to B_{ZZ}$ smoothly as $s_Z \to 0$. In addition the PBU and LHU formulas will reduce to a projected Broyden update of B_{YZ} and a projected PSB update of B_{YY}.

4.2. Root Selection and Rejection

In the previous section we showed that the cubic equation has three real roots only when $\omega \leq 0$. When $\omega=0$ we have the three real roots

$$\alpha_1^0 = 2A$$
$$\alpha_2^0 = \alpha_3^0 = -A \quad ,$$

where α^0 indicates roots corresponding to $\omega=0$. In the case $\omega<0$ we have the three roots

$$\alpha_1^- = 2\,real(A)$$
$$\alpha_2^- = -real(A) - \sqrt{3}\,imag(A)$$
$$\alpha_3^- = -real(A) + \sqrt{3}\,imag(A) \quad ,$$

where α^- indicates $\omega<0$. We showed that when $s_Y \cong 0$, α_1^- and α_2^- are very close to the roots α_{BFGS} for the standard BFGS update while α_3^- approaches the fictitious root, zero. Clearly, if our update is to approximate the standard BFGS update, it is important to eliminate the fictitious root and choose either α_1^- or α_2^- which are near α_{BFGS}. Note that α_i^- smoothly approaches α_i^0 for $i = 1,2,3$ as $\omega \to 0_-$ since $imag(A) \to 0$ as $\omega \to 0_-$. Consequently, when $\omega \leq 0$, choosing one of the roots α_1 or α_2 will guarantee that the selected root corresponds to α_{BFGS} when $\nu \cong 0$ and insures that the fictitious root is eliminated.

Which of the two roots, α_1 or α_2, should we select? Unlike the standard BFGS update, the roots do not give the same update. Also, there is no apparent theoretical justification for selecting one root over the other. After extensive experimentation the following selection rule was adopted. First, since we wish the roots to be near the standard BFGS roots, when the quantity $s_Z{}^T r_Z$ is positive and s_Y is small relative to s_Z we select the root whose absolute value is nearest to $+(s_Z{}^T r_Z / s_Z{}^T B_{ZZ} s_Z)^{1/2}$. We consider s_Y small relative to s_Z when $|\nu/\eta| < 0.10$ since the Taylor expansions (4.6) indicate that the cubic roots will be near the roots α_{BFGS} in this case. When $s_Z{}^T r_Z < 0$, we fall back on the same logic used to select the root in the standard BFGS update: choose the root which makes $\|E\|_2 = \|u\|_2 \|w\|_2$ as small as possible. Here, $w_{LHU}, w_{PBU} = \alpha L^T s_Z$ and

$$u_{LHU} = \frac{r_Z - \alpha B_{ZZ} s_Z}{s_Y{}^T s_Y + \alpha^2 s_Z{}^T B_{ZZ} s_Z}$$
$$u_{PBU} = \frac{2(r_Z - \alpha B_{ZZ} s_Z - \gamma s_Z)}{s_Y{}^T s_Y + 2\alpha^2 s_Z{}^T B_{ZZ} s_Z}$$
$$\gamma = \frac{s_Y{}^T r_Y - s_Y{}^T s_Y (\alpha-1)/2\alpha}{s_Y{}^T s_Y + s^T s} \quad .$$

When $\omega>0$ we saw in the previous section that only one real root exists; thus no selection of a root is required. However, when $s_Y \ll s_Z$ and $s_Z{}^T y_Z \leq 0$ and $\omega>0$, ill-conditioning may occur. To see this, we note that (4.7) yields $s_Z{}^T B_{ZZ}^+ s_Z = \alpha^2 s_Z{}^T B_{ZZ} s_Z$ which implies that as $\alpha \to 0$, B_{ZZ}^+ approaches singularity. To avoid ill-conditioning, it may be necessary to reject the single real cubic root and skip the update. Fortunately we can predict this situation fairly accurately by estimating the condition number of the new matrix B_{ZZ}^+ before updating. Defining the condition number of a matrix A as $K(A) \doteq \lambda_{max}/\lambda_{min}$ we estimate λ_{max} and λ_{min}, the largest and smallest eigenvalues of A. By definition, the largest and smallest eigenvalues of A satisfy the relations $\max\{\lambda \mid Ax=\lambda x,\ \|x\|=1\}$ and $\min\{\lambda \mid Ax=\lambda x,\ \|x\|=1\}$ respectively. Thus, for all x we have $x^T Ax / x^T x \leq \lambda_{max}$ and $x^T Ax / x^T x \geq \lambda_{min}$.

First consider the quantity $s_Z{}^T B_{ZZ}^{+} s_Z = \alpha^2 s_Z{}^T B_{ZZ} s_Z$. For small α, $s_Z{}^T B_{ZZ}^{+} s_Z / s_Z{}^T s_Z$ is clearly a measure of the smallest eigenvalue of B_{ZZ}^{+}. Thus, we estimate λ_{min} by

$$\tilde{\lambda}_{min} \doteq \frac{s_Z{}^T B_{ZZ}^{+} s_Z}{||s_Z||^2} = \frac{\alpha^2 s_Z{}^T B_{ZZ} s_Z}{||s_Z||^2} \geq \lambda_{min}$$

where $\lambda_{max}=\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{n-t}=\lambda_{min}>0$ are the eigenvalues of B_{ZZ}^{+}.

We must also estimate the largest eigenvalue. Using $B_{ZZ}=LDL^T$ and the definition of λ_{max} we have for all x

$$\lambda_{max} = ||B_{ZZ}^{+}|| \geq \frac{x^T B_{ZZ}^{+} x}{x^T x} = \frac{1}{||x||^2} x^T (LDL^T + \frac{pp^T}{p^T s_Z} - \frac{B_{ZZ} s_Z s_Z{}^T B_{ZZ}}{s_Z{}^T B_{ZZ} s_Z}) x$$

and thus, for the best estimate, we seek x to maximize the right hand side. If d_i is the largest entry in the diagonal matrix D, we can estimate the desired x by l_{*i}: column i of the matrix L. Thus,

$$\tilde{\lambda}_{max} \doteq \frac{1}{||l_{*i}||^2} l_{*i}^T (LDL^T + \frac{pp^T}{p^T s_Z} - \frac{B_{ZZ} s_Z s_Z{}^T B_{ZZ}}{s_Z{}^T B_{ZZ} s_Z}) l_{*i} \leq \lambda_{max}$$

is an estimate of the largest eigenvalue of B_{ZZ}^{+}.

Now, define $\tilde{K}$ as an estimate of the condition number,

$$\tilde{K} \doteq \frac{\tilde{\lambda}_{max}}{\tilde{\lambda}_{min}} \leq \frac{\lambda_{max}}{\lambda_{min}} \doteq K(B_{ZZ}^{+}) \quad .$$

Since we are most concerned about the case $\alpha \to 0$ and B_{ZZ}^{+} nearly singular, the estimate of λ_{min} is the most important component of the condition number estimate. Fortunately the estimate of $\tilde{\lambda}_{min}$ will be fairly accurate when B_{ZZ}^{+} is almost singular since the singularity occurs in a direction nearly parallel to s_Z. This can easily be seen by considering (4.7) which yields $B_{ZZ}^{+} s_Z = p$. Since $p \to 0$ as $\alpha \to 0$, B_{ZZ}^{+} approaches singularity in the direction s_Z.

To confirm that $\tilde{K}$ is a reasonable estimate, it was compared to a condition number estimate computed from the Cholesky decomposition of B_{ZZ}^{+}. This condition number estimator was based on the routine DCHCO from the LINPACK (Dongarra, et al., 1979) code. At all iterations, our estimate $\tilde{K}$ was found to be within an order of magnitude of the estimate obtained from the Cholesky decomposition. We concluded that $\tilde{K}$ could be used as a predictor of ill-conditioning in our numerical experiments. An alternative approach we considered was to use the more accurate LINPACK routine to estimate the condition number after performing the update and then reset the Hessian approximation when the condition number of the new approximation was too large. However, this procedure would have required additional storage as well as the extra computational effort to compute the LINPACK estimate and thus was rejected in favor of the simpler predictor, $\tilde{K}$.

5. Computational Testing

The LHU and PBU formulas were tested by implementing them in two algorithms based on the projected form of the SQP step. The local algorithm uses a modified SQP step without line search or trust region. The semi-global algorithm includes a line search on the augmented Lagrangian merit function

$$L_a(x, \lambda^{LS}, \sigma) \doteq f(x) - c(x)^T \lambda^{LS} + \frac{\sigma}{2} c(x)^T c(x)$$

to insure that the Goldstein-Armijo conditions are satisfied at each iteration. Thus, we have

$$L_a(x_{k+1},\lambda_k^{LS},\sigma)<L_a(x_k,\lambda_k^{LS},\sigma)$$

for each iteration when σ is sufficiently large. However this approach can not guarantee global convergence. At the end of each iteration the Lagrange multipliers are recomputed using a least-squares estimate. Unfortunately, we can not guarantee

$$L_a(x_{k+1},\lambda_{k+1}^{LS},\sigma)<L_a(x_{k+1},\lambda_k^{LS},\sigma)$$

so that L_a may not decrease. Near the solution, however, the multipliers will change very little and we can anticipate a monotonic decrease of the augmented Lagrangian.

In the tests, the Lower Half and Partitioned BFGS updates were compared with what could be considered the standard projected-BFGS update formula used in nonlinearly constrained optimization. The Coleman and Conn (1984) formula discussed earlier is of this type. These updates approximate only the null-space projection of the full Hessian matrix using an extra midpoint gradient evaluation to compute y_Z, the change in projected gradients due to the step in the null space. Letting B_{ZZ} represent the current approximation to the null-space projection of the Hessian, we define the BFGS-PM (Projected Midpoint) update by

$$\begin{aligned} s_Z &= Z^T(h+v) = Z^T h \\ y_Z &= Z^T(\nabla L(x+h+v)-\nabla L(x+v)) \quad . \\ B_{ZZ}^{+} &= B_{ZZ} + \frac{y_Z y_Z{}^T}{s_Z{}^T y_Z} - \frac{B_{ZZ} s_Z s_Z{}^T B_{ZZ}}{s_Z{}^T B_{ZZ} s_Z} \quad , \end{aligned} \qquad \text{(BFGS-PM)}$$

where, again, h is a step in the null space and v is a step in the range space. With the extra gradient evaluated at $x+v$, $s_Z{}^T y_Z>0$ can be expected in a neighborhood of the solution. By collecting additional information at each iteration we can also expect more accurate Hessian approximations. Since this update should perform well near x^*, we will use it as a standard to measure the new updates. To maintain positivity, we skipped the BFGS-PM update when $s_Z{}^T y_Z<\varepsilon_{mach}^{1/2}$.

In this section we present the two algorithms, the initialization procedures, the test problems studied, and several tests including comparisons with the results of Nocedal and Overton (1985).

5.1. Two Algorithms for Testing the LHU and PBU Updates

The two algorithms shown below differ mainly in two features: the definition of the null-space step $h=Zs_Z$ and the use of a line-search procedure in the semi-global algorithm. The semi-global algorithm uses the SQP x-step and the least squares estimate of λ. A line search is also performed on the full step $s=h+v$.

SEMI-GLOBAL ALGORITHM

INITIALIZE:

$k \leftarrow 0$.

Choose σ, M_σ, x_0, B_{YY}^0, B_{YZ}^0, B_{ZY}^0, B_{ZZ}^0.

Evaluate f_0, ∇f_0, c_0, A_0.

Factor $A_0=[Y_0 \;\; Z_0][R_0^T \;\; 0^T]^T$.

Evaluate $\lambda_0^{LS}=R_0^{-1}Y_0^T\nabla f_0$, $L_a(x_0,\lambda_0^{LS},\sigma)$.

DO WHILE ($||Z_k^T\nabla f_k||>\varepsilon_{stop}$ or $||c_k||>\varepsilon_{stop}$)

$v_k \leftarrow Y_k s_Y^k = -Y_k R_k^{-T} c_k$

$h_k \leftarrow Z_k s_Z^k = -Z_k (B_{ZZ}^k)^{-1}(Z_k^T \nabla f_k + B_{ZY}^k s_Y^k)$

$s_k \leftarrow h_k + v_k$

CALL LINESEARCH(σ, τ_k, x_k, s_k, L_a^k) .

DO WHILE (Armijo-Goldstein conditions on L_a not satisfied by τ_k)

$\sigma \leftarrow \sigma M_\sigma$

CALL LINESEARCH(σ, τ_k, x_k, s_k, L_a^k) .

END

$x_{k+1} \leftarrow x_k + \tau_k s_k$

Evaluate f_{k+1}, ∇f_{k+1}, c_{k+1}, A_{k+1}.

Factor $A_{k+1} = [Y_{k+1} \;\; Z_{k+1}][R_{k+1}^T \;\; 0^T]^T$.

Evaluate λ_{k+1}^{LS}, $L_a(x_{k+1}, \lambda_{k+1}^{LS}, \sigma)$.

$B^{k+1} \leftarrow U(B^k, x_k, v_k, h_k)$

$k \leftarrow k+1$.

END

Note that the local approach has no linesearch and uses a modified h-step which ignores the change in gradients due to the step $v = Ys_Y$.

LOCAL ALGORITHM

INITIALIZE:

$k \leftarrow 0$.

Choose x_0, B_{YY}^0, B_{YZ}^0, B_{ZY}^0, B_{ZZ}^0.

Evaluate f_0, ∇f_0, c_0, A_0.

Factor $A_0 = [Y_0 \;\; Z_0][R_0^T \;\; 0^T]^T$.

Evaluate $\lambda_0^{LS} = R_0^{-1} Y_0^T \nabla f_0$.

DO WHILE ($||Z_k^T \nabla f_k|| > \varepsilon_{stop}$ or $||c_k|| > \varepsilon_{stop}$)

$v_k \leftarrow Y_k s_Y^k = -Y_k R_k^{-T} c_k$

$h_k \leftarrow Z_k s_Z^k = -Z_k (B_{ZZ}^k)^{-1} Z_k^T \nabla L_k$

$s_k \leftarrow h_k + v_k$

$x_{k+1} \leftarrow x_k + s_k$

Evaluate f_{k+1}, ∇f_{k+1}, c_{k+1}, A_{k+1}.

Factor $A_{k+1} = [Y_{k+1} \;\; Z_{k+1}][R_{k+1}^T \;\; 0^T]^T$.

Evaluate λ_{k+1}^{LS}.

$B^{k+1} \leftarrow U(B^k, x_k, v_k, h_k)$

$k \leftarrow k+1$.

END

5.2. Some Algorithmic Details

The Hessian approximation is stored in projected form and we use Gill, Golub, Murray, and Saunders' (1974) method to update the Cholesky factorization of $B_{ZZ} = LDL^T$ at each iteration. The formulas presented earlier are used to define either the LHU or PBU update after solving the appropriate cubic equation. The methods discussed in the previous section were used to select a root when more than one real root exists and also predict ill-conditioning of B_{ZZ}. In all computational testing, the LHU and PBU updates were skipped if the condition number estimate for B_{ZZ}^{+}, $\tilde{K}$, exceeded 10^5. When it was necessary to skip the PBU update, the matrices B_{YY} and $B_{YZ}=B_{ZY}{}^T$ were updated using the Modified PBU described in Section 3.

For the LHU and PBU we initialize $B_{ZY}^0 \leftarrow 0$ and $B_{YZ}^0 \leftarrow 0$. The initial approximation $B_{YY}^0 \leftarrow I_{t\times t}$ is used for the PBU. For most of the testing the first step is computed using $B_{ZZ}^0 \leftarrow I_{(n-t)\times(n-t)}$. When using $B_{ZZ}^0=I$ and $B_{YY}^0=I$ we obtain better estimates of B_{ZZ} and B_{YY} by scaling the identity matrices at the end of the first iteration, before the first update. Consider scaling $B_{ZZ}^0=\rho_{11}I$ and $B_{ZZ}^0=\rho_{22}I$. Since $Y^T\nabla^2LZ$ and $Z^T\nabla^2LY$ are non-zero they satisfy

$$\begin{bmatrix} Y^T\nabla^2LY & Y^T\nabla^2LZ \\ Z^T\nabla^2LY & Z^T\nabla^2LZ \end{bmatrix} \begin{Bmatrix} s_Y \\ s_Z \end{Bmatrix} = \begin{Bmatrix} y_Y \\ y_Z \end{Bmatrix} + O(\|s\|^2)$$

where y_Y and y_Z are projections of the gradient change $\nabla L(x+Ys_Y+Zs_Z)-\nabla L(x)$. Using an extra gradient evaluation at the point x_0+Ys_Y we define

$$\bar{y}_Y \doteq Y^T(\nabla L(x_0+Ys_Y)-\nabla L(x_0))$$
$$\bar{y}_Z \doteq Z^T(\nabla L(x_0+Ys_Y+Zs_Z)-\nabla L(x_0+Ys_Y))$$

and approximate

$$Y^T\nabla^2LYs_Y \cong y_Y - Y^T\nabla^2LZs_Z \cong \bar{y}_Y$$
$$Z^T\nabla^2LZs_Z \cong y_Z - Z^T\nabla^2LYs_Y \cong \bar{y}_Z \quad .$$

Since we want B_{YY} and B_{ZZ} to approximate $Y^T\nabla^2LY$ and $Z^T\nabla^2LZ$ in the directions s_Y and s_Z respectively, we set $\rho_{11} \leftarrow s_Y{}^T\bar{y}_Y / s_Y{}^Ts_Y$ and $\rho_{22} \leftarrow |s_Z{}^T\bar{y}_Z| / s_Z{}^Ts_Z$ which yields

$$s_Y{}^T\rho_{11}B_{YY}^0s_Y = s_Y{}^T\bar{y}_Y$$
$$s_Z{}^T\rho_{22}B_{ZZ}^0s_Z = |s_Z{}^T\bar{y}_Z|$$

as desired. When using the BFGS-PM update, the extra gradient at $x+Ys_Y$ is already available since it is required to compute the update. For the LHU and PBU algorithms, the extra gradient for the initial scaling increases the total number of gradient evaluations by one. Because the additional work is small, this improved scaling was implemented in both the semi-global and local algorithms.

Some additional details are required to describe the computation of the step $h=-Z(B_{ZZ})^{-1}(Z^T\nabla f+B_{ZY}s_Y)$ in the semi-global algorithm. When using the LHU and PBU, the matrix B_{ZY} is readily available to compute the additional term $B_{ZY}s_Y$. When using the BFGS-PM update we must approximate the term $B_{ZY}s_Y$. Since the extra gradient, $\nabla L(x+Ys_Y)$, is required to compute the BFGS-PM update we also use it to approximate the vector $B_{ZY}s_Y$ with $Z^T(\nabla L(x+Ys_Y)-\nabla L(x))$.

Finally, the line-search procedure used in the semi-global algorithm was a test version of the routine CVSRCH supplied by J. J. Moré at Argonne National Laboratory. CVSRCH uses a combination of a cubic and a quadratic fit to find the safeguarded step which satisfies the sufficient decrease and curvature conditions of the Armijo-Goldstein criteria.

5.3. The Test Problems

The problems briefly described below were used to test the new updates in both the local and semi-global algorithms. Detailed descriptions of these problems may be found in Fenyes (1987) or in earlier sources. The first four problems were taken from Nocedal and Overton (1985). The next three are from Wright (1976) and Rosen and Suzuki (1965) and the problems labeled HSxx are from Hock and Schittkowski (1981).

TEST PROBLEMS				
PROBLEM	N	T	OBJECTIVE	CONSTRAINTS
NO1	2	1	POLY O(3)	POLY O(2)
NO2	3	2	POLY O(2)	POLY O(2)
NO3	5	3	EXP + POLY O(6)	POLY O(3)
NO4	5	3	POLY O(4)	POLY O(3)
WRIGHT 8	9	6	POLY O(2)	POLY O(2)
WRIGHT 9	5	2	TRIG + POLY O(9)	POLY O(3)
RS	4	2	POLY O(2)	POLY O(2)
HS27	3	1	POLY O(4)	POLY O(2)
HS53	5	3	POLY O(2)	POLY O(1)
HS60	3	1	POLY O(4)	POLY O(4)
HS61	3	2	POLY O(2)	POLY O(2)
HS77	5	2	POLY O(6)	TRIG + POLY O(6)
HS78	5	3	POLY O(5)	POLY O(3)
HS79	5	3	POLY O(4)	POLY O(3)
HS100	7	2	POLY O(6)	POLY O(4)

Here, N is the number of variables, T the number of constraints, and OBJECTIVE and CONSTRAINTS give a brief description of the objective and constraint functions. We use POLY O(n) to represent a polynomial of order n and EXP and TRIG to represent exponential and trigonometric functions respectively.

5.4. Comparison with Nocedal and Overton's Method

Nocedal and Overton's (1985) local algorithm, C7, uses the same step, s, and gradient change, y, as our local algorithm given above. They use a projected BFGS update formula, updating only if the step satisfies

$$||v_k|| < \frac{\eta}{(k+1)^{1+\nu}} ||h_k||$$

where they take ν=0.01 and η=1.0. Although Nocedal and Overton's results are quite local, we felt it would be informative to compare our results with theirs. We selected five of their test problems, P1, P2, P3, P4 and problem 100 from Hock and Schittkowski. In the listing above, these are referred to as problems NO1, NO2, NO3, NO4, and HS100 respectively. Using Nocedal and Overton's starting points, we ran the local algorithm above, updating with the LHU, the PBU, and the BFGS-PM update. Like Nocedal and Overton, we initialize the LHU, PBU, and BFGS-PM methods by setting B_{ZZ}^0 to a finite difference estimate of $Z^T\nabla^2 LZ$. We insure positivity of this initial approximation by factoring it with the modified Cholesky decomposition of Gill and Murray (1974). For the LHU, and PBU we set $B_{YZ}^0 = B_{ZY}^0 = 0$ and $B_{YY}^0 = I$. The matrices B_{YY}, B_{YZ}, and B_{ZY} are not used in the BFGS-PM method. For this test we set $\varepsilon_{stop} = 10^{-8}$.

The results are shown in Table 5.1 where the BFGS-PM and Nocedal and Overton methods are listed as BFGS and NO/C7 respectively. The columns labels are:

TABLE 5.1

LOCAL ALGORITHM: B_{22} BY FINITE DIFFERENCE								
PROBLEM	UPDATE	NIT	NGD	NUS	NSY	NT1	NT2	NT3
NO1	LHU	5	6	0	0	1	0	4
	PBU	6	7	0	0	1	0	4
	BFGS	5	11	0	0	1	0	4
	NO/C7	5	6	1	-	1	-	-
NO2	LHU	6	7	0	1	2	2	4
	PBU	6	7	0	1	3	2	4
	BFGS	5	11	0	0	3	2	4
	NO/C7	6	7	3	-	3	-	-
NO3	LHU	5	6	0	0	3	2	3
	PBU	6	7	0	0	3	1	4
	BFGS	4	9	0	0	2	1	2
	NO/C7	5	6	1	-	1	-	-
NO4	LHU	9	10	0	0	0	0	4
	PBU	10	11	0	0	1	0	5
	BFGS	8	17	0	0	1	0	5
	NO/C7	8	9	0	-	0	-	-
HS100	LHU	11	12	0	0	0	0	4
	PBU	12	13	0	0	1	0	6
	BFGS	11	23	0	0	1	0	5
	NO/C7	12	13	0	-	0	-	-

NIT - Number of iterations
NGD - Number of times the gradients were evaluated
NUS - Number of times the update was skipped
NSY - Number of times $s^T y < 0$
NT1 - Number of times the Nocedal and Overton test failed with $\eta=1.0$
NT2 - Number of times the Nocedal and Overton test failed with $\eta=2.0$
NT3 - Number of times the Nocedal and Overton test failed with $\eta=0.1$

The count NGD does not include the gradients required to obtain the initial finite difference approximation of B_{ZZ}. Also, since an extra gradient is required to check convergence the last iteration, NGD=2(NIT)+1 for the BFGS-PM, and NGD=NIT+1 for all other methods.

All three updates achieve convergence in approximately the same number of iterations. The BFGS-PM method requires slightly fewer iterations but, as we would expect, this is achieved at the expense of nearly twice as many gradient evaluations. It is also interesting to note that even though the starting points used were very near the solutions, the method NO/C7 skips updates in three of the problems while the other methods are able to update at every iteration. In problem NO2, the algorithm NO/C7 seems to perform well even though Nocedal and Overton are only able to perform two of the five possible updates. Skipping an update apparently has little effect in these problems because the initial finite difference Hessian approximation is taken near the solution.

5.5. Further Testing of the Local and Global Algorithms

The results in Table 5.1 are based on single starting points quite near the solution. Since the LHU and PBU were formulated to maintain a positive definite Hessian approximation even far from the solution, additional testing was performed to study the updates with a variety of starting points. A total of 540 starting points were randomly selected: 36 starting points for each of the 15 problems listed above. Twelve points were chosen from each of the full, range, or null spaces determined at the solution: three starting points at each of the distances 0.01, 0.10, 1.00 and 5.00 from the solution. By considering many problems with many starting points we obtain an unbiased view of the performance of the LHU, PBU, and the BFGS-PM updates when used in either the local or semi-global algorithm. In all cases we initialize B_{YY} and B_{ZZ} to scaled identity matrices and set $B_{YZ}=B_{ZY}=0$. We note that for the BFGS-PM update we must have $s_Z^T y_Z>0$ in order to update. For the LHU and PBU it may still be possible to update even when $s_Z^T y_Z<0$.

Due to space considerations, we cannot possibly give all the computational results for the local and semi-global algorithms. Many of these results are presented in Fenyes (1987) and all the results we obtained have been analyzed and summarized below. The starting points were divided into two groups: NEAR, for the points at radius 0.01 and 0.10, and FAR, for the points at radius 1.00 and 5.00. First, we determined the number of iterations and gradient evaluations required for convergence of each algorithm when using the BFGS-PM, LHU, and PBU formulas from each starting point. Using SAS (SAS Institute Inc., 1982) we analyzed our results as discussed below.

The averages, grouped by starting distance from the solution, are shown in Figures 5.1 for the local algorithm and 5.2 for the semi-global algorithm. They give a measure of the relative efficiencies for each update formula, but do not indicate if the differences are significant. To study these differences, we paired the tests by starting point and averaged, within groups, the quantities

$$\begin{aligned} &NIT_k(UPDT_i)-NIT_k(UPDT_j) \\ &NGD_k(UPDT_i)-NGD_k(UPDT_j)\quad . \end{aligned} \tag{5.1}$$

Here, $NIT_k(UPDT_i)$ indicates the number of iterations required for convergence of update i from starting point k while i and j represent any of the BFGS-PM, LHU, or PBU formulas. The paired t-test was used to condense the voluminous information and determine if the differences between updates were statistically significant. The paired t-statistics are presented in Tables 5.2, 5.3, and 5.4. Averages of the differences (5.1) are shown in the AVG columns. Under the null hypothesis that the means for the three updates were equal, we computed the significance levels shown in the column labelled SIG LVL for the alternate hypotheses $\overline{NIT}(UPDT_i)<\overline{NIT}(UPDT_j)$ or, depending on the sign of the average difference, $\overline{NIT}(UPDT_i)>\overline{NIT}(UPDT_j)$. Here, we use the overbar to indicate an average. For values of

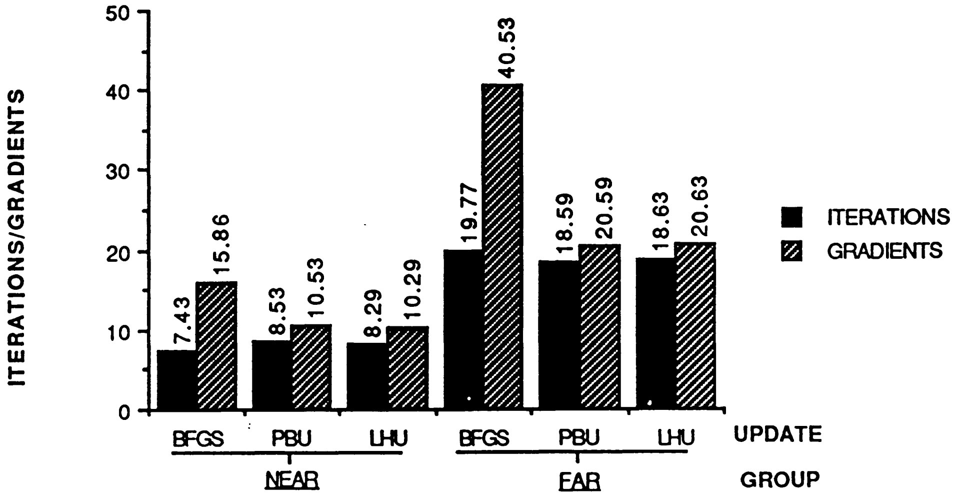

FIGURE 5.1 - LOCAL ALGORITHM

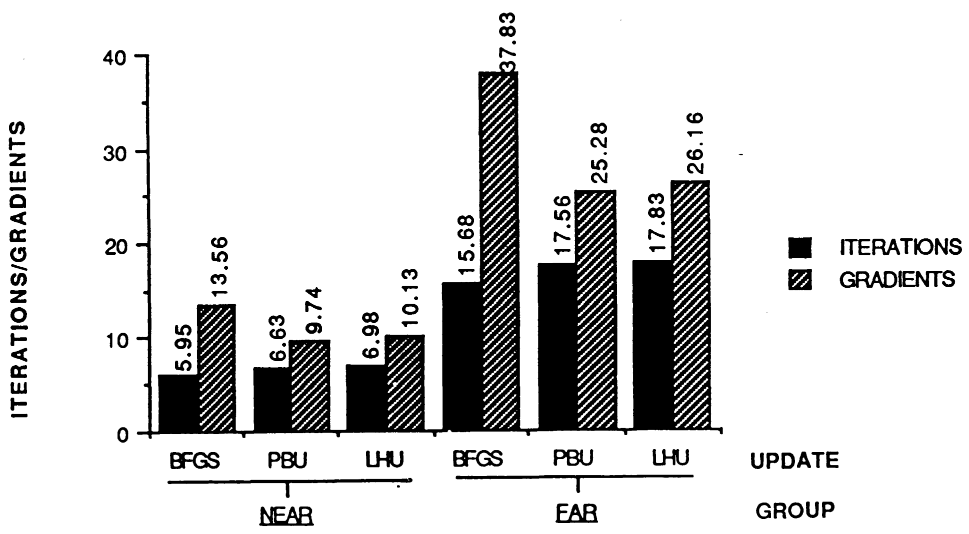

FIGURE 5.2 - SEMI-GLOBAL ALGORITHM

SIG LVL less than 0.01 we concluded that the null hypothesis was false and that the two methods were statistically different. In other words for SIG LVL less than 0.01, when $AVG>0$

[$AVG<0$] we could then conclude that $\overline{NIT}(UPDT_i) > \overline{NIT}(UPDT_j)$ [$\overline{NIT}(UPDT_i) < \overline{NIT}(UPDT_j)$]. For SIG LVL greater than 0.01 we concluded that the update methods were, on average, the same.

Let us first consider Table 5.2 which compares the number of iterations required for convergence of the BFGS-PM with the LHU and PBU formulas.

TABLE 5.2

COMPARISON WITH BFGS-PM: ITERATIONS				
ALGO	UPDT	POINTS	AVG(BFGS-UPDT)	SIG LVL
LOCAL	LHU	NEAR	-0.86	0.00005
		FAR	+0.30	0.36970
	PBU	NEAR	-1.09	0.00005
		FAR	+0.16	0.39890
S-GLO	LHU	NEAR	-1.03	0.00005
		FAR	-1.65	0.00690
	PBU	NEAR	-0.68	0.00005
		FAR	-1.65	0.00450

We would expect the BFGS-PM to require fewer iterations for convergence since it uses the additional gradient information at the midpoints. However, Table 5.2 shows this advantage is small. For starting points near the solution, the BFGS-PM formula has about a single iteration advantage over both the LHU and PBU formulas. For starting points far from the solution, the slight advantage of the BFGS-PM formula is maintained in the semi-global algorithm, but not for the local algorithm. No statistically significant difference can be demonstrated for the BFGS-PM compared to the LHU and PBU when using the local algorithm from starting points far from the solution.

Table 5.3 compares the number of gradients required for convergence in the BFGS-PM against the LHU and PBU. As we would expect, the BFGS-PM requires many more gradient evaluations than either the LHU or PBU formulas and the differences are statistically significant. In fact, we could show, at a significance level of 0.01, that the BFGS-PM formula requires more than 13 [8] extra gradients than the LHU or PBU when using the local [semi-global] algorithm far from the solution.

In Table 5.4 we look for differences between the two new updates. The only statistically significant difference is the lower iteration count for the PBU formula when using the semi-global algorithm near the solution.

TABLE 5.3

COMPARISON WITH BFGS-PM: GRADIENTS				
ALGO	UPDT	POINTS	AVG(BFGS-UPDT)	SIG LVL
LOCAL	LHU	NEAR	+5.57	0.00005
		FAR	+18.0	0.00005
	PBU	NEAR	+5.33	0.00005
		FAR	+17.71	0.00005
S-GLO	LHU	NEAR	+3.43	0.00005
		FAR	+13.11	0.00005
	PBU	NEAR	+3.82	0.00005
		FAR	+12.95	0.00005

TABLE 5.4

COMPARISON BETWEEN LHU AND PBU				
ALGO	COUNT	POINTS	AVG(LHU-PBU)	SIG LVL
LOCAL	NIT	NEAR	-0.23	0.20385
		FAR	-0.25	0.32200
	NGD	NEAR	-0.23	0.20385
		FAR	-0.25	0.32200
S-GLO	NIT	NEAR	+0.35	0.00010
		FAR	-0.06	0.46125
	NGD	NEAR	+0.39	0.16040
		FAR	-0.26	0.41480

6. Discussion

It is clear that the LHU and the PBU formulas have several desirable algorithmic features. First, no extra midpoint gradient evaluations are needed to insure positivity of B_{ZZ}. Thus, these updates are very efficient. Secondly, since we do not need to satisfy a condition of the type $s^T y>0$ when $s_Y \neq 0$ and $s_Z \neq 0$, they will allow us to update our Hessian approximation at nearly every iteration, even when we are far from the solution. Finally, Corollaries 3.2, 3.3, 3.5, and 3.6 show that the PBU and LHU formulas reduce to projected forms of the Broyden, PSB, and BFGS updates in the special cases $s_Y=0$ or $s_Z=0$. This is a particularly nice property of the LHU and PBU formulas.

Theorems 3.1 and 3.4 show that these updates are well defined when both $s_Y \neq 0$ and $s_Z \neq 0$ and that the formulas will keep B_{ZZ} positive definite. In some sense this is counter intuitive. Suppose we have taken a step which lies mostly in the null space, in other words $s_Y \cong 0$, and we have $s_Z{}^T y_Z<0$. In this situation the standard projected BFGS formula fails to keep B_{ZZ} positive definite and it must therefore be skipped. In contrast, although we have seen no positive curvature in the null space, we can still update with either the PBU or LHU and maintain positivity of B_{ZZ}. How can this be? Comparing the projected BFGS formula with the PBU we can resolve this apparent paradox. First consider the quasi-Newton equations for both formulas. For the standard BFGS we satisfy an equation of the form

$$B_{ZZ}^{+} s_Z = y_Z \quad .$$

If $s_Z{}^T y_Z = s_Z{}^T B_{ZZ}^{+} s_Z < 0$ then clearly B_{ZZ}^{+} is not positive definite and the update can not be performed. However, for the PBU we have the quasi-Newton equation

$$B_{ZZ}^{+} s_Z = y_Z - B_{ZY}^{+} s_Y \quad .$$

Here, again, $s_Z{}^T B_{ZZ}^{+} s_Z$ must be positive if B_{ZZ}^{+} is to be positive definite. But this implies $s_Z{}^T y_Z - s_Z{}^T B_{ZY}^{+} s_Y > 0$. In this case, when $s_Z{}^T y_Z < 0$ we can force $s_Z{}^T B_{ZZ}^{+} s_Z > 0$ by choosing B_{ZY}^{+} to satisfy $s_Z{}^T B_{ZY}^{+} s_Z < s_Z{}^T y_Z$. In some sense, B_{ZY}^{+} serves as an additional degree of freedom to maintain the positivity of B_{ZZ}^{+}. Of course, only certain choices of B_{ZY}^{+} will minimize the matrix norm and satisfy the quasi-Newton equation - these choices are determined by the roots, α, of the cubic equation. Even when $s_Y \cong 0$ and $s_Z{}^T y_Z < 0$ this additional freedom allows us to keep B_{ZZ}^{+} positive definite.

In Section 4 we analyzed the cubic roots and discussed choosing a root when three real roots are available. The most difficult situation occurs when $s_Y << s_Z$. Let us summarize our results for the three sub cases. We have shown that if $\omega < 0$ the LHU and PBU formulas smoothly reduce to the standard projected BFGS update of B_{ZZ} while $B_{YY}^{+} \rightarrow B_{YY}$ and B_{ZY}^{+} reduces to a projected Broyden. This insures a reasonable approximation of the Hessian. We also showed that $\omega = 0$ or $\omega > 0$ may yield the real root $\alpha \cong 0$. However, since $s_Z{}^T B_{ZZ}^{+} s_Z = \alpha^2 s_Z{}^T B_{ZZ} s_Z$, updating with $\alpha \cong 0$ may lead to ill-conditioning. When only the single real root $\alpha \cong 0$ is available it may be necessary to skip the new updates. Fortunately, under reasonable assumptions, we may argue that $s_Y << s_Z$ and $\omega \geq 0$ will occur very infrequently. Consider the following. Although it is possible for $s_Y << s_Z$ at any step of the algorithm, it will most likely occur when we are near the solution. Thus, we can expect both $s_Z{}^T y_Z > 0$ and $\omega < 0$ since we have assumed the null-space projection of the Hessian to be positive definite near x^*. We conclude that when $s_Y << s_Z$ both new updates will correspond to the standard projected BFGS update as desired.

One desirable property of the BFGS and DFP updates is their invariance to variable scalings of the type $\bar{x} = Dx$ where D is a diagonal scaling matrix. This insures that a change of units or an arbitrary scaling of the variables will not affect the updates or the algorithmic performance. The Broyden and PSB updates do not have this scale invariance. Unfortunately, the LHU and PBU updates are not scale invariant. However, the LHU update will be nearly scale invariant if $s_Y{}^T s_Y$ is small relative to $s_Z{}^T B_{ZZ} s_Z$.

The numerical results of Section 5 are quite encouraging. The comparisons with the method of Nocedal and Overton have shown that our update formula leads to more frequent updates, even near the solution, and thus may be more practical. Finally, the statistical analysis using a large number of starting points showed that the PBU and LHU are quite competitive with the standard projected BFGS update. The LHU and PBU converge in about the same number of iterations as the projected BFGS, however they require significantly fewer gradient evaluations.

Acknowledgment We wish to thank J. J. Moré for the use of the CVSRCH linesearch procedure which he wrote with D. J. Thuente at Argonne National Laboratories.

BIBLIOGRAPHY

Abramowitz, M. and Stegun, I. A. (1964) **Handbook of Mathematical Functions,** Applied Mathematics Series, Vol. 55, National Bureau of Standards, Washington, DC. (Reprinted 1968, Dover Publications, New York, NY)

Armijo, L. (1966) *Minimization of Functions Having Lipschitz-Continuous First Partial Derivatives,* Pacific J. Math., 16, pp. 1-3.

Biggs, M. C. (1972) *Constrained Minimization Using Recursive Equality Quadratic Programming,* in **Numerical Methods for Non-linear Optimization**, F. A. Lootsma, ed., Academic Press, London, New York, pp. 411-428.

Biggs, M. C. (1975) *Constrained Minimization Using Recursive Quadratic Programming: Some Alternative Subproblem Formulations,* in **Towards Global Optimization**, L. C. W. Dixon and G. P. Szegö, eds., North-Holland Publishing, Amsterdam, pp. 341-349.

Bogen, R. (1983) **MACSYMA Reference Manual,** Symbolics, Inc., Cambridge, MA.

Brent, R. P. (1973) **Algorithms for Minimization without Derivatives,** Prentice-Hall, Inc., Englewood Cliffs, NJ.

Coleman, T. F. and Conn, A. R. (1984) *On the Local Convergence of a Quasi-Newton Method for the Nonlinear Programming Problem,* SIAM J. Numer. Anal., 21, no. 4, pp. 755-769.

Debreu, G. (1952) *Definite and Semidefinite Quadratic Forms,* Econometrica, 20, pp. 295-300.

Dennis, J. E., Jr. and Schnabel, R. B. (1979) *Least Change Secant Updates for Quasi-Newton Methods,* SIAM Review, 21, no. 4, pp. 443-459.

Dongarra, J. J., Bunch, J. R., Moler, C. B., and Stewart, G. W. (1979) **LINPACK Users' Guide,** SIAM Publications, Philadelphia, PA.

Fenyes, P. A. (1987) *Partitioned Quasi-Newton Methods for Nonlinear Equality Constrained Optimization*, Ph.D. Thesis, Cornell University, Ithaca, NY.

Fontecilla, R. (1987) *Local Convergence of Secant Methods for Nonlinear Constrained Optimization (Revised Manuscript)*, Department of Computer Science, Institute for Physical Science and Technology, U. of Maryland, College Park, MD 20742.

Gabay, D. (1982) *Reduced Quasi-Newton Methods with Feasibility Improvement for Nonlinearly Constrained Optimization*, Math. Programming Study, 16, pp. 18-44.

Garcia Palomares, U. M. and Mangasarian, O. L. (1976) *Superlinearly Convergent Quasi-Newton Algorithms for Nonlinearly Constrained Optimization Problems*, Math. Programming, 11, pp. 1-13.

Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H. (1985) *Model Building and Practcal Aspects of Nonlinear Programming*, in **Computational Mathematical Programming**, K. Schittkowski, ed., NATO ASI Series, Vol.F15, Springer-Verlag, Berlin, pp. 209-247.

Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H. (1986) *User's Guide for NPSOL (Version 4.0): A Fortran Package for Nonlinear Programming*, Technical Report SOL 86-2, Systems Optimization Laboratory, Stanford Univ., Stanford, CA.

Glad, S. T. (1979) *Properties of Updating Methods for the Multipliers in Augmented Lagrangians*, J. Optim. Theory Appl., 28, pp. 135-156.

Goldstein, A. A. (1967) **Constructive Real Analysis**, Harper & Row, New York, NY.

Gurwitz, C. B. (1986) *Sequential Quadratic Programming Methods Based on Approximating a Projected Hessian Matrix*, Tech. Report #219, Department of Computer Science, Courant Institute of Mathematical Sciences, New York, NY.

Han, S.-P. (1976) *Superlinearly Convergent Variable Metric Algorithms for General Nonlinear Programming Problems*, Math. Programming, 11, pp. 263-282.

Hestenes, M. R. (1969) *Multiplier and Gradient Methods*, J. Optim. Theory Appl., 4, pp. 303-320.

Hock, W. and Schittkowski, K. (1981) **Test Examples for Nonlinear Programming Codes**, Lecture Notes in Economics and Mathematical Systems, No. 187, Springer-Verlag, New York, NY.

Householder, A. S. (1970) **The Numerical Treatment of a Single Nonlinear Equation,** McGraw-Hill, New York, NY.

Mayne, D. Q. and Polak, E. (1982) *A Superlinearly Convergent Algorithm for Constrained Optimization Problems,* Math. Programming Study, 16, pp. 45-61.

Murray, W. (1969) *An Algorthm for Constrained Minimization,* in **Optimisation**, R. Fletcher, ed., Academic Press, London, pp. 247-258.

Murray, W. and Wright, M. H. (1978) *Projected Lagrangian Methods Based on the Trajectories of Penalty and Barrier Functions,* Report SOL 78-23, Dept. Operations Research, Stanford Univ., Stanford, CA.

Nocedal, J. and Overton, M. (1985) *Projected Hessian Updating Algorithms for Nonlinearly Constrained Optimization,* SIAM J. Numer. Anal., 22, pp. 821-850.

Powell, M. J. D. (1978) *A Fast Algorithm for Nonlinearly Constrained Optimization Calculations,* in **Numerical Analysis. Proceedings of the Biennial Conference Held at Dundee, June 1977**, G. A. Watson, ed., Lecture Notes in Mathematics, Vol. 630, Springer-Verlag, New York, NY, pp. 144-157.

Powell, M. J. D. (1985) *The Performance of Two Subroutines for Constrained Optimization on Some Difficult Test Problems,* in **Numerical Optimization 1984**, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., SIAM, Philadelphia, PA, pp. 160-177.

Powell, M. J. D. and Yuan, Y. (1986) *A Recursive Quadratic Programming Algorithm that Uses Differentiable Exact Penalty Functions,* Math. Programming, 35, pp. 265-278.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1986) **Numerical Recipes: The Art of Scientific Computing,** Cambridge Univ. Press, New York, NY.

Rosen, J. B. and Suzuki, S. (1965) *Construction of Nonlinear Programming Test Problems,* Comm. ACM, 8, p. 113.

SAS Institute Inc. (1982) **SAS User's Guide: Basics 1982 Edition,** SAS Institute Inc., Cary, NY.

Schittkowski, K. (1981) *The Nonlinear Programming Method of Wilson, Han, and Powell with an Augmented Lagrangian Type Line Search Function,* Numer. Math., 38, pp. 83-114.

Schittkowski, K. (1984) *A Unified Nonlinear Programming Theory for Penalty, Multiplier, SQP and GRG Methods,* in **System Modelling and Optimization, Proc. 11th IFIP Conference, Copenhagen, Denmark, July 25-29, 1983**, P. Thoft-Christensen, ed., Springer-Verlag, Berlin, pp. 299-310.

Tanabe, K. (1981) *Feasibility-improving Gradient-acute-projection Methods: A Unified Approach to Nonlinear Programming.*, Lecture Notes in Num. Appl. Anal., 3, pp. 57-76.

Tapia, R. A. (1977) *Diagonalized Multiplier Methods and Quasi-Newton Methods for Constrained Optimization,* J. Optim. Theory Appl., 22, no. 2, pp. 135-194.

Tapia, R. A. (1978) *Quasi-Newton Methods for Equality Constrained Optimization: Equivalence of Existing Methods and a New Implementation,* in **Nonlinear Programming 3**, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, New York, NY.

Womersley, R. S. and Fletcher, R. (1982) *An Algorithm for Composite Nonsmooth Optimization Problems,* Report NA/60, Dept. of Mathematics, Univ. of Dundee, Dundee, Scotland.

Wright, M. H. (1976) *Numerical Methods for Nonlinearly Constrained Optimization,* Ph. D. Thesis, Stanford Univ., Stanford, CA.